

Filed: August 15, 2025

Filed on behalf of Meta Platforms, Inc.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.,
Petitioner,

v.

DIALECT, LLC,
Patent Owner

Case No. IPR2025-01336
U.S. Patent No. 9,734,825

**Petition for *Inter Partes* Review of
U.S. Patent No. 9,734,825**

TABLE OF CONTENTS

I. INTRODUCTION	1
II. IDENTIFICATION OF CHALLENGES.....	2
III. BACKGROUND	2
A. '825 Patent.....	2
B. Prosecution History	3
IV. THE PRIOR ART.....	4
A. Ross (EX1005)	4
B. Ittycheriah (EX1007).....	7
C. Hartono (EX1006).....	9
D. Baker (EX1008)	12
E. Glover (EX1011).....	14
V. LEVEL OF ORDINARY SKILL IN THE ART.....	16
VI. CLAIM CONSTRUCTION	17
VII. GROUND 1.....	17
A. Motivation to Combine	17
B. Reasonable Expectation of Success	19
C. Independent Claim 5	21
1. Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:”	21

2. 5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;” 23
3. 5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;” 25
4. 5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;” 29
5. 5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;” 30
6. 5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;” 32
7. 5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;” 35
8. 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;” 36

9.	5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”	38
D.	Dependent Claims 6-7	40
1.	Claim 6: “further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.”	40
2.	Claim 7: “wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”	41
VIII. GROUND 2		42
A.	Motivation to Combine	42
B.	Reasonable Expectation of Success	44
C.	Independent Claim 5	46
1.	Preamble through 5[f]	46
2.	5[g]: “creating...”	46
3.	5[h]: “sending...”	46
D.	Dependent Claims 6-8	48
1.	Claim 6: “further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.”	48
2.	Claim 7: “wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”	48

3. Claim 8: “wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.” 48

IX. GROUND 3.....50

- A. Motivation to Combine 50
- B. Reasonable Expectation of Success 52
- C. Independent Claim 5 53
1. Preamble.....53
2. 5[a]: “recognizing...” 53
3. 5[b]: “receiving...” 58
4. 5[c]: “determining...a score...” 59
5. 5[d]: “determining...a domain...” 62
6. 5[e]: “selecting...” 63
7. 5[f]: “providing...” 66
8. 5[g]: “creating...” 67
9. 5[h]: “sending...” 67
- D. Dependent Claims 69
1. Claim 6: “...generating...” 69
2. Claim 7: “...comprises a plurality...” 70
3. Claim 8: “...create...and send...” 71

X. GROUND 4.....	73
A. Motivation to Combine	73
B. Reasonable Expectation of Success	75
C. Claim 8: “...create...and send...”	76
XI. SECONDARY CONSIDERATIONS.....	77
XII. MANDATORY NOTICES.....	77
A. Real Parties-in-Interest	77
B. Related Matters.....	78
C. Notice of Counsel and Service Information.....	78
D. Fee for <i>Inter Partes</i> Review	79
E. Grounds for Standing	80
XIII. CONCLUSION	80

TABLE OF AUTHORITIES

Page(s)

Cases

<i>Dialect LLC v. Meta Platforms, Inc.</i> , No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025).....	10, 81
<i>Dialect, LLC v. Microsoft Corp.</i> , Case No. 2:24-vv-01067 (E.D. Tex.).....	81

Statutes

35 U.S.C. § 103	2
-----------------------	---

Other Authorities

37 C.F.R. § 42.10(b)	82
37 C.F.R. § 42.15(a).....	83
37 C.F.R. §42.24(a).....	87
37 C.F.R. § 42.24(d)	87
37 C.F.R. § 42.100(b)	17
37 C.F.R. § 42.104(a).....	83

TABLE OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 9,734,825 to Kennewick, et al. (“the ’825 Patent”)
1002	File History of U.S. Patent No. 9,734,825
1003	Declaration of Daniel Lopresti (“Lopresti”)
1004	Curriculum Vitae of Daniel Lopresti
1005	U.S. Patent No. 7,085,723 to Ross et al. (“Ross”)
1006	WO 00/11571 to Hartono et al. (“Hartono”)
1007	U.S. Patent No. 5,937,383 to Ittycheriah et al. (“Ittycheriah”)
1008	U.S. Patent No. 5,680,511 to Baker et al. (“Baker”)
1009	U.S. Patent No. 7,249,018 to Ross et al. (“Ross018”)
1010	U.S. Patent No. 6,636,848 to Aridor et al. (“Aridor”)
1011	U.S. Patent Application Publication No. 2002/1065860 to Glover et al. (“Glover”)
1012	U.S. Patent No. 7,127,402 to Ross et al. (“Ross402”)
1013	Reserved
1014	Reserved

Exhibit No.	Description
1015	Appendix E – Infringement Contentions for U.S. Patent No. 9,734,825, <i>Dialect LLC v. Meta Platforms, Inc.</i> , No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025)
1016	U.S. Provisional Patent Application No. 60/261372 to Ross et al.
1017	IBM® ViaVoice® for Windows® Personal Edition, User’s Guide, Release 10 (August 2002)
1018	Excerpts from IBM ViaVoice™ 98 Home Edition User Guide (July 1998)
1019	Excerpts from Frederick Jelinek, <i>Statistical Methods for Speech Recognition</i> , The MIT Press (1997)
1020	Michael McTear, <i>Spoken Dialogue Technology: Enabling the Conversational User Interface</i> , 34 ACM Computing Surveys No. 1, pp. 90-169 (Mar. 2002)
1021	Liam Comerford, et al., <i>The IBM Personal Speech Assistant</i> , IBM Watson Research Center, IEEE (2001)
1022	Eric J. Glover, et al., <i>Architecture of a Metasearch Engine that Supports User Information Needs</i> , In International Conference on Information and Knowledge Management, Proceedings, pp. 210-216, ACM (1999)
1023	Dragon Naturally Speaking® 5, User’s Guide (August 2000)
1024	Definitions of “asynchronous” and “token” from Microsoft Computer Dictionary, Fourth Edition, Microsoft Corporation (1999)

Exhibit No.	Description
1025	Definition of “dictionary” from Webster’s Ninth New Collegiate Dictionary, Merriam-Webster Inc. (1990)
1026	Definition of “dictionary” from The American Heritage Dictionary of The English Language, Houghton Mifflin Co. (1992)

I. INTRODUCTION

Meta Platforms, Inc. (“Petitioner”) respectfully requests *inter partes* review of claim 5-8 of U.S. Patent No. 7,398,825 (EX1001, “’825 Patent”).

Natural language speech recognition methods have been used for decades. *See generally* Lopresti ¶¶39-25. By the mid-1990s, several companies including IBM and Dragon Systems, sold speech recognition systems off-the-shelf to process user-generated natural language speech into machine-readable text to be used by autonomous computer applications. These systems used statistical probabilities, language modeling, and context-recognition procedures to process speech. Importantly, because these systems employed autonomous applications to respond to natural language queries, those applications operated asynchronously from the main programming flow of speech recognition and parsing.

The ’825 Patent describes speech recognition methods that process natural language queries among various application programs. Although the Examiner repeatedly rejected the original claims of the ’825 Patent, they were allowed after the applicant added limitations requiring a domain agent to send requests asynchronously. But asynchronous programming was commonplace by 2002, as it was well-known that domain agents could operate independent of one another.

Accordingly, Petitioner respectfully requests the Board institute review and find all challenged claims of the ’825 Patent unpatentable.

II. IDENTIFICATION OF CHALLENGES

Ground	Claims	Statute	Prior Art
1	5-7	Pre-AIA 35 U.S.C. § 103	Ross (EX1005), Ittycheriah (EX1007)
2	5-8		Ross (EX1005), Ittycheriah (EX1007), Glover (EX1011)
3	5-8		Hartono (EX1006), Baker (EX1008)
4	8		Hartono (EX1006), Baker (EX1008), Glover (EX1011)

For the purposes of this Petition only, Petitioner applies the earliest possible priority date—June 3, 2002—for the '825 Patent. Petitioner reserves the right to contest priority in other proceedings.

III. BACKGROUND

A. '825 Patent

The '825 Patent describes well-known systems and methods that “determine a domain for a user generated natural language utterance based on the content and context of the natural language utterance.” EX1001, Abstract. For example, the '825 Patent recognizes that “[m]uch work covering multiple methods has been done in the fields of natural language processing and speech recognition,” and “[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications.” *Id.*, 1:56-60. The '825 Patent further recognizes that

“[n]atural language processing has previously been applied to the parsing of speech queries.” *Id.*, 1:60-61.

The '825 Patent claims recite nothing more than a combination of these well-known systems and methods. At a high level, the '825 Patent recites the steps of using a “speech recognition engine” to recognize one or more words in user-generated speech. *Id.*, 27:9-28. It next recites using the one or more words with associated prior probabilities or fuzzy possibilities to determine possible contexts for the received utterance and provides a query or command based on that context. *Id.*, 27:28-28:11. The '825 Patent lastly recites sending the query or command to a local or external information source to be executed. *Id.*, 28:11-17. To perform several of these steps, the '825 Patent teaches using an “active domain agent of a plurality of autonomous executable domain agents,” which it describes as being “complete, convenient and re-distributable packages or modules for each application area,” i.e., “everything needed to extend or modify the functionality of the system. *Id.*, 11:58-63.

B. Prosecution History

The original claims of the '825 Patent were repeatedly rejected as being unpatentable over U.S. Patent No. 6,523,061 to Halverson et al. (“Halverson”). Notably, even after several amendments, the Examiner stated “the Applicants’

current claims are simply too broad given what is known in this field of the prior art.” EX1002, 220.

Finally, on March 22, 2017, the Applicant amended the claims to add “wherein each of the selected at least one of the plurality of domain agents is configured to create one or more queries based on the at least one query and/or command and send the one or more queries in an asynchronous manner to one or more local or external information sources.” *Id.*, 231. The Applicant admitted that Halverson “describes agents as being able to issue requests to facilitator 600 for additional assistance,” but emphasized that “Halverson does not disclose or suggest that the requests are sent in an asynchronous manner.” *Id.*, 235.

On April 12, 2017, the Examiner allowed the claims, stating that the prior art of record does “not disclose or reasonably suggest sending queries in an asynchronous manner.” *Id.*, 253.

IV. THE PRIOR ART

A. Ross (EX1005)

Ross is a U.S. patent assigned to IBM entitled “System and Method for Determining Utterance Context in Multi-Context Speech Application” filed on August 16, 2001 and issued on August 1, 2006. Ross is prior art at least under

pre-AIA §102(a) and (e). EX1005, Cover. Ross discloses each limitation of independent claim 5 except for the details of the *speech engine*.

Ross discloses a “speech center” that “coordinates speech services for a number of speech-enabled applications performing on a computer system,” including the “task of deciding the target application for a particular utterance by a user.” EX1005, Abstract, 2:3-9, Fig. 1; Lopresti ¶67. Upon receiving a speech input, Ross uses a speech engine to convert spoken commands (i.e., requests) “into text strings that can be utilized by software applications,” i.e., a “speech representation.” EX1005, 6:62-7:2. Ross recognizes that by 2001, speech engines were already well known and commercially available including ViaVoice™ from IBM and NaturallySpeaking Professional from Dragon Systems. EX1005, 1:16-22; *see generally* EX1017, EX1018, EX1025; Lopresti ¶68.

Ross describes a “context manager” receiving the speech representation from such a “speech engine” and proceeds to “select a context 70 for a speech enabled application 26 by checking the speech representation against contexts 70.” EX1005, 7:19-23; Lopresti ¶69. Ross ranks the contexts based on an “access characteristic,” such as the “recency of relevant access to the context,” as shown in Figure 4:

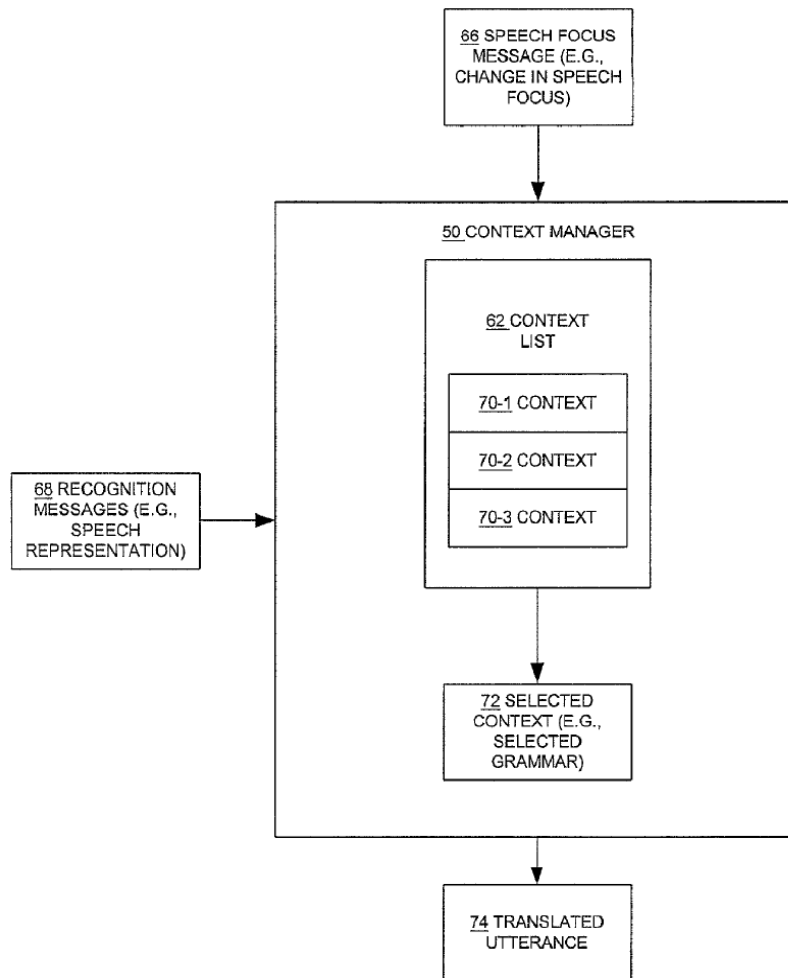


FIG. 4

EX1005, Fig. 4, 2:47-49, 3:1-3, 6:47-61, 8:46-9:13. Consequently, Ross maintains a set of prior probabilities of possible contexts “biased to send commands to the most recently accessed application,” that is continuously being “reordered.” EX1005, 8:47-53; *see also* 5:58-61; Lopresti ¶70. Ross then uses “the grammar associated with each context 70 to determine if the [speech representation] is capable of being

processed by the grammar,” and selects the context based on this matching (i.e., relevance scoring). EX1005, 7:23-26; Lopresti ¶71. Ross translates the speech representation “according to the selected context 72 and directs the translation to the script engine,” the script engine then runs the script against the target application, thereby invoking it to perform the command and present the results to the user. EX1005, 7:27-28, 6:15-18, Fig.4; Lopresti ¶72.

B. Ittycheriah (EX1007)

Ittycheriah is a U.S. patent entitled “Apparatus and Methods for Speech Recognition Including Individual or Speaker Class Dependent Decoding History Caches for Fast Word Acceptance or Rejection” filed on June 4, 1997, and issued on August 10, 1999. Ittycheriah is thus prior art under pre-AIA §102(a), (b), and (e). EX1007, Cover. Ittycheriah details a *speech engine*.

As shown in Figure 1, Ittycheriah discloses a speech recognition engine for “performing speech recognition on speech segments frequently input by a user”:

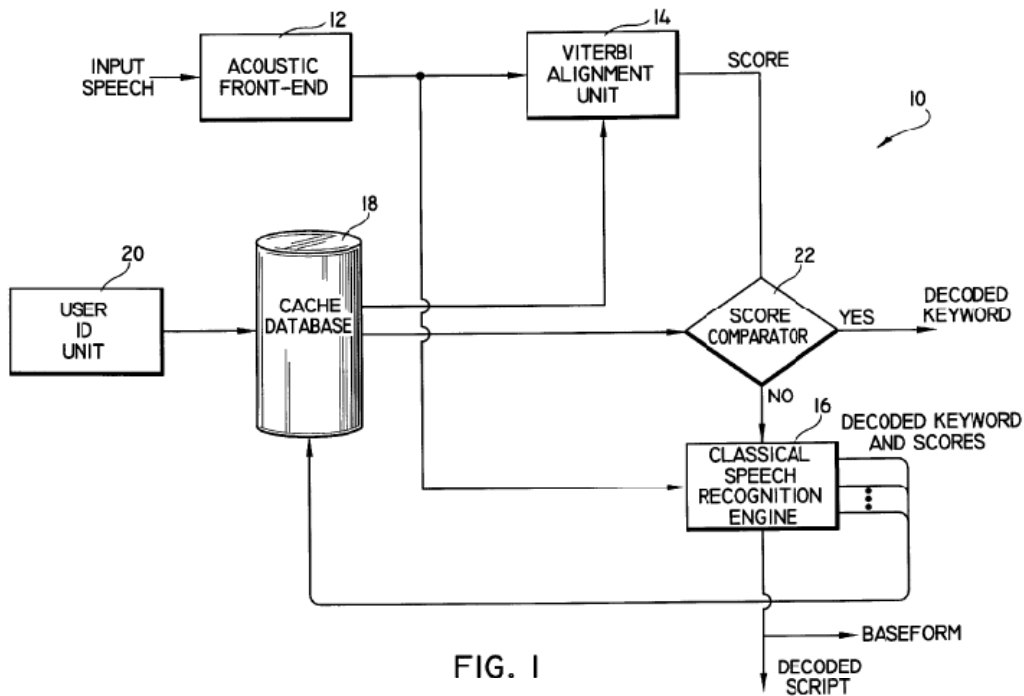


FIG. 1

EX1007, Fig. 1, Abstract, 1:11-15, 1:51-54; Lopresti ¶¶74. Ittycheriah accomplishes this by “storing data in a cache and utilizing such data in order to quickly accept or reject keywords.” EX1007, 2:59-63. Ittycheriah’s cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” EX1007, 2:63-66; Lopresti ¶¶75-76.

Ittycheriah uses a set of probabilities to perform speech recognition, explaining that once speech is input into its system, it subjects the acoustic features of speech to “probabilistic matching technique using “Hidden Markov models (HMMs)” that “provide different scores for different hypotheses” each based on probabilities. EX1007, 3:56-4:4; Lopresti ¶¶77-83. Ittycheriah then compares the

“resulting scores” to “scores which were previously stored in the cache database 18 pertaining to the particular user are presented to the score comparator 22 wherein the scores from unit 14 are compared to the stored scores.” EX1007, 4:33-43. “If the new scores are within a margin of, the same as, or larger than the stored scores corresponding to one of the decoded keywords in the cache, then the keyword is accepted,” and the decoded keyword is outputted from the score comparator. EX1007, 4:43-48. If the new scores are “not within a margin of error, or are less than any of the stored scores, then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine,” from which a decoded script (i.e., a keyword) is outputted. EX1007, 4:48-53.

C. Hartono (EX1006)

Hartono is a PCT application entitled “Adaptive Natural Language Interface,” with an International Publication date of March 2, 2000. EX1006, Cover. Hartono is thus prior art under §102(a) and (b). Hartono discloses each limitation of independent claim 5 except for the details of the *speech engine*.

Hartono discloses a “method for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.” EX1006, 1:20-23; Lopresti ¶¶85-87. To that end, Hartono describes a

“natural language agent adapted to receive and interpret natural language user input and to output a command,” and an “application agent adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program. EX1006, 4:20-25. Hartono illustrates its use of various application agents in Figure 1:

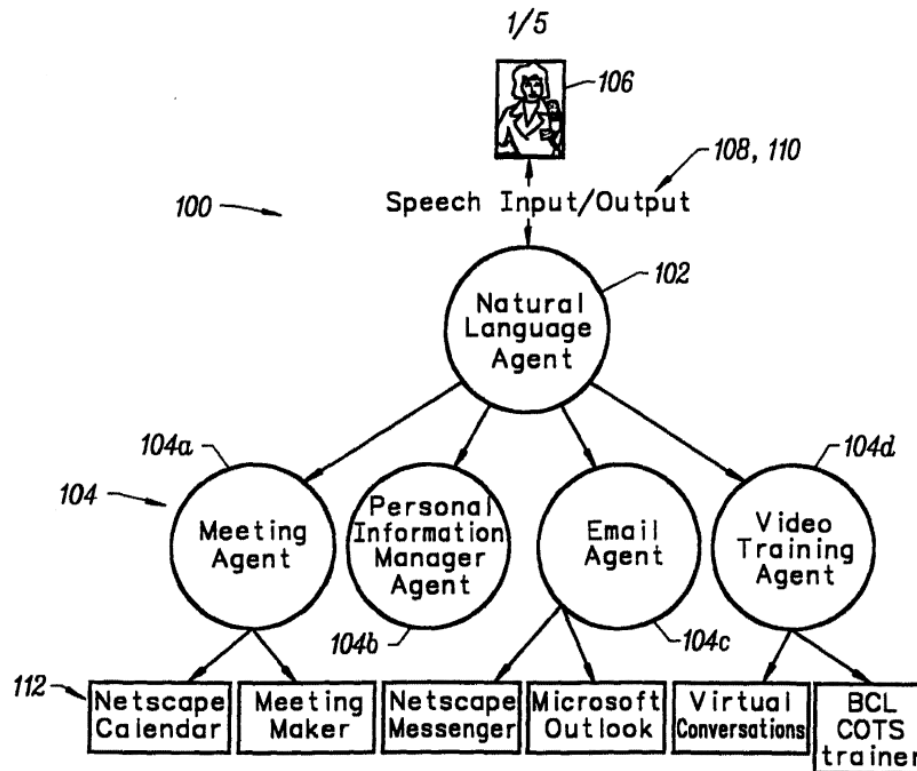


FIG. 1

EX1006, Fig. 1, 7:6-9 (“[W]ith a modular distributed agent architecture, the system and method of the present invention is extendable to multiple applications and is scalable to large sets of networked computer systems.”).

Hartono initially processes input speech using “[a]ny suitable off the shelf speech recognition system.” EX1006, 11:4-12. After receiving the recognized speech, Hartono discloses using a “natural language syntactic parser” to “parse natural language computer commands” from the recognized, for example, converting “the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as the properties of the object.” EX1006, 12:11-13:10. Hartono then takes the parsed natural language and “maps it into a generic concept frame used to invoke the appropriate application method,” as shown in Table I:

TABLE I		
ACTION-CONCEPT	TOPIC CONCEPT	APPLICATION-CONTEXT
Show	Email	Email application
Show	Email address	Address book application
Delete	Email	Email application
Show	General help	Natural language agent

EX1006, Table I. To complete this mapping, Hartono’s semantic interpretation engine uses a “reverse grammar generation mechanism” that “includes a list or vector for each word and corresponding probabilities for each word in the list.” EX1006, 14:10-20. For example, Hartono describes the words “I,” “eye,” or “aye,” having probabilities of 80%, 15%, and 5%, respectively such that the word with the highest probability is most likely to be suitable for the given context. EX1006, 15:4-

12. Using both the action-concept (verb) and topic-concepts (noun/object), Hartono determines the context of the natural language request by “determin[ing] which task agent should handle the request.” EX1006, 15:13-19; Lopresti ¶¶88-93.

Hartono communicates with various task agents (also called “application class agents”), which “works with a single class of applications...that have similar conceptual operations.” EX1006, 23:15-23, 28:1-3 (“Each task agent thus serves as the common user interface for the corresponding class of applications under the assumption that each application within the class accomplishes the same or generally overlapping set of tasks.”). Based on the determined context, Hartono “invokes a routine that sends a set of request to the task application wrapper,” which is “written in one of the platform specific macro languages.” EX1006, 24:14-25:3. The command within the natural language request is then executed by the selected application. EX1006, 4:20-25, 5:4-8, 5:11-15, 29:21-22, 42:11-12; Lopresti ¶¶94-96.

D. Baker (EX1008)

Baker is a U.S. Patent entitled “Systems and Methods for Word Recognition” filed on June 7, 1995 and issued on October 21, 1997. Baker is thus prior art under pre-AIA §102(a), (b), and (e). EX1008, Cover. Baker details a *speech engine*.

Baker describes “detect[ing] words within portions of speech signals and

compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” EX1008, 18:13-16; Lopresti ¶¶98-102. Upon receiving speech, Baker compares the speech to “acoustic models of known words, or sequence of acoustic models of known words.” EX1008, 18:12-16. “This comparison leads to a probability signal that indicates the certainty that the detected word is substantially acoustically similar to the word associated with the acoustic model.” EX1008, 18:16-19. Words with a probability signal indicating a “high degree of certainty” are deemed recognized (called “context words”). EX1008, 18:16-19. But words with a “low degree of certainty” indicate that the word has not been confidently identified (called “current word”). EX1008, 18:22-24.

For those unidentified words, Baker maintains a “choice list 18” that “store[s] the words...[that are] acoustically similar to the current ambiguous word 38 being identified. EX1008,18:27-30; Lopresti ¶¶103-108. Each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” representing “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” EX1008, 18:30-35, 20:30-46, 22:55-64. Baker also discloses generating “word lists” that “have a semantic association with the choice word 20.” EX1008, 14:57-69. Baker “perform[s] a matching function to calculate the number of matched words” between those word

recognized by the speech analysis unit 66 with high confidence (the “context words”) with the words in the “word list” (the “passage words”). EX1008, 21:19-47. Using the relative frequency of matches between the “context words” and “passage words,” Baker calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” EX1008, 21:19-47, 21:49-22:6, Figs, 2-4.

To complete its speech recognition process, Baker’s generates a “rank signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized as a function of both the “context matching and acoustic model matching” probabilities. EX1008, 23:22-27; Lopresti ¶¶109-111. Baker selects the choice word with the highest rank signal and supplants the unrecognized word in the text string with the choice word as a way of recognizing the received utterance. EX1008, 12:54-62, 17:23-27 (“[T]he word recognition system 10 can select the choice word 20 with the highest probability signal 28 or rank signal 30 to substitute for the current word 38 currently being recognized.”).

E. Glover (EX1011)

Glover is a U.S. patent application publication entitled “Selective Retrieval Metasearch Engine” filed on June 29, 2001, and published on November 7, 2002. Glover is prior art at least under pre-AIA §102(a) and (e). Glover discloses a

metasearch engine that teaches sending *duplicate queries* to multiple search engines (*different information sources*) as recited in dependent claim 8.

Glover's metasearch engine "takes user queries and submits them to multiple underlying search engines, and combines the results into a single interface." EX1011, ¶¶35, 47 ("A retrieve search engine results step 52 for sending queries to search engines or databases and retrieving the results from the search engines or databases[.]"); *see id.*, ¶3 ("A metasearch engine accepts a search query, and send the query (possibly transformed) to one or more regular search engines, and collects and processes the responses from the regular search engines in order to present a list of documents to the user."); Lopresti ¶¶113-114. Glover's system is illustrated in Figure 2:

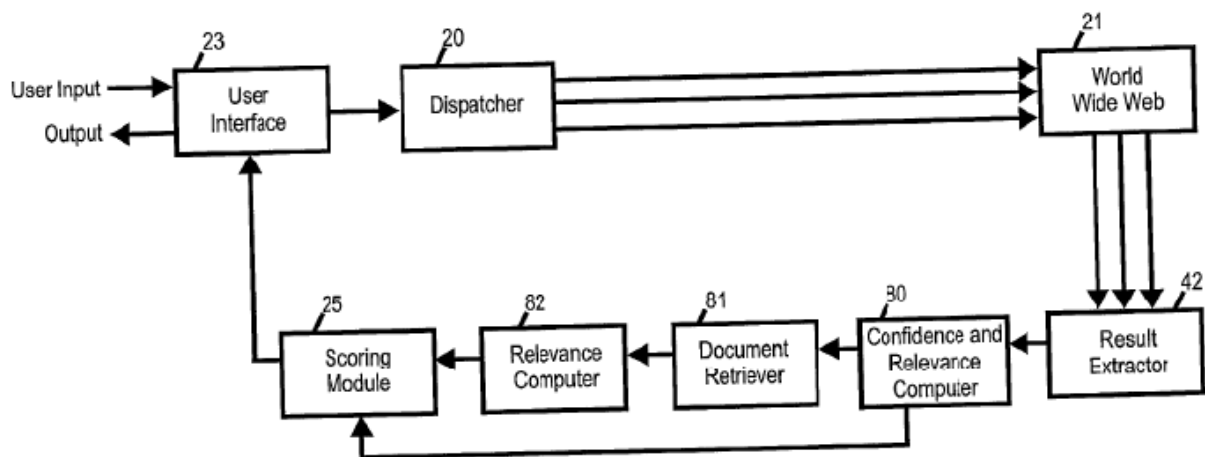


FIG. 8

EX1011, Fig. 8; *see also id.*, Fig. 2, ¶¶36-38.

To facilitate fast, accurate searching, Glover describes using a “dispatcher” (item 20) to “make[] the primary search decisions for a metasearch engine,” including “which search engines to query, and how to query each source.” EX1011, ¶¶38, 39, Fig. 3; Lopresti ¶¶115-116. Glover explains that “some of the steps” of its claimed method “may be performed in a parallel,” including “sending queries 51 to and retrieving results 52 from search engines.” EX1011, ¶82.

V. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) in June 2002 would have had at least a bachelor’s degree in electrical engineering, computer science, computer engineering, or an equivalent, and two years of relevant experience involving computer science fundamentals, including natural language processing, and speech recognition and transcription. Additional experience could substitute for less education, and additional education could likewise substitute for less experience. Lopresti ¶¶59-62.

This Petition does not turn on this precise definition, and the challenged claims would be unpatentable from the perspective of any reasonable person of ordinary skill in the art at the relevant time. Lopresti ¶¶63-65.

VI. CLAIM CONSTRUCTION

The Board construes the claims “using the claim construction standard that would be used” in district courts. 37 C.F.R. § 42.100(b). This Petition establishes the prior art meets each claim limitation under any reasonable construction.¹ Therefore, no express construction is necessary.

VII. GROUND 1

A. Motivation to Combine

A POSITA would have been motivated to use Ittycheriah’s speech recognition method to implement the details of Ross’s “speech engine,” because Ross teaches using other off-the-shelf IBM speech recognition systems such as Ittycheriah. Lopresti ¶¶119-122. Ross uses a “speech engine” to “convert speech into text strings that can be utilized by software applications on a computer.” EX1005, 1:7-22. Ross does not disclose the details of the “speech engine,” but instead recognizes that off-the-shelf speech engines could be used to convert speech to text. EX1005, 1:16-21, 4:37-38 (“In a preferred embodiment, the speech engine 22 is ViaVoice™ from IBM®.”). Indeed, the provisional patent application from which Ross depends, states that “[w]hile we certainly expect to make use of IBM’s ViaVoice speech

¹ Petitioner reserves the right to argue alternative constructions in other proceedings, including that the claims are indefinite where such a defense is available.

engine, we would be able to easily replace it with a better alternative if one came along.” EX1016, 3. Thus, a POSITA would have been motivated to look for and rely on available speech recognition methods to use as Ross’s “speech engine,” including Ittycheriah’s speech recognition engine (also developed by IBM). Lopresti ¶122.

A POSITA would have also been motivated to implement Ross’s speech engine using Ittycheriah to quickly complete the speech recognition process with its cache database of recently used keywords and probability scores, advancing an objective of Ross to avoid time and expense. EX1005, 2:14-16; Lopresti ¶123.

Moreover, a POSITA would have used Ittycheriah’s speech recognition system as Ross’s speech engine because Ittycheriah completes speech recognition faster than classical speech engines with its cache database that stores recently-recognized keywords and associated probabilities. Lopresti ¶¶124-125; *see* Section IV.B (Ittycheriah Overview). It is desirable for speech-enabled requests and commands (like those inputted into Ross) to be processed quickly. Thus, a POSITA would have found Ittycheriah’s cache database desirable to “reduce the amount of hypotheses to consider in the decoding process.” EX1007, 2:13-19, 2:58-63.

B. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in combining the teachings of Ross and Ittycheriah. Lopresti ¶¶126-129. A POSITA would have used Ittycheriah's speech recognition process to implement the details of Ross's speech engine 22, as shown in Figure 2, which Ross teaches could have been any commercially-available speech recognition engine:

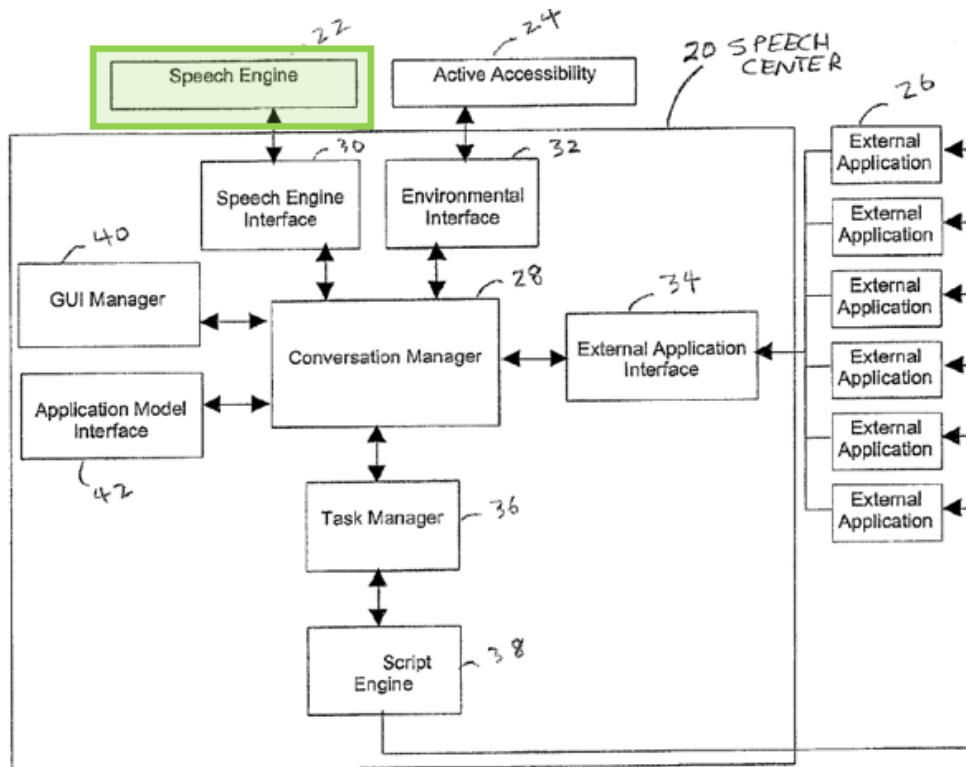


Fig. 2

EX1005, Fig. 2 (annotated)

This is particularly true since IBM developed both Ross and Ittycheriah. EX1005, [73]; EX1007, [73].

Indeed, a POSITA would have understood that Ittycheriah's speech recognition system would have been compatible with Ross, since Ittycheriah discloses that decoded keywords and scripts could be used to direct speech requests to external speech-enabled applications to perform their respective automatic functions, as in Ross. EX1007, 4:63-5:3; Lopresti ¶128.

A POSITA would have also had the skill to implement and reasonable expectation of success in achieving the combination because it would have involved a combination of known technologies (speech recognition and processing natural language requests) according to known methods (speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of inputting natural language speech and outputting a text string that can invoke various speech-enabled applications on a computer. Lopresti ¶129.

C. Independent Claim 5

- 1. Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:”**

To the extent limiting, Ross discloses the preamble. Lopresti ¶¶131-139. As discussed in Section IV.A (Ross Overview), Ross discloses “determining a speech-enabled application to receive a spoken utterance in a multi-context speech enabled environment.” EX1005, 2:43-46, Abstract. The spoken utterance is *a user generated natural language speech utterance*, and Ross’s speech center includes *a method for responding to the utterance*. EX1005, 5:34-38; Lopresti ¶¶132-133.

Ross discloses several examples of responding to natural language requests. *See, e.g.*, EX1005, 2:25-28, 2:31-36, 8:25-33 (“[I]f the speech center **20** hears a phrase such as ‘print the first message’ or ‘print the first appointment,’ the context manager **50** can readily figure out the intended target application **26** for the uttered sentence.”). As an example, Ross describes that when “the user asks whether the user has any new messages,” the user “receives an answer supplied by a message or mail application.” EX1005, 2:28-31. A POSITA would have understood that these voice commands are *natural language* because they use phrases in human-understandable language. Lopresti ¶¶134-136.

Ross's methods *responding to a user generated natural language speech utterance* is illustrated in Figure 1, showing a processor receiving "input spoken utterance 14," which is a "voice command or other speech input from a user of a computer system" and producing output 16 in response:

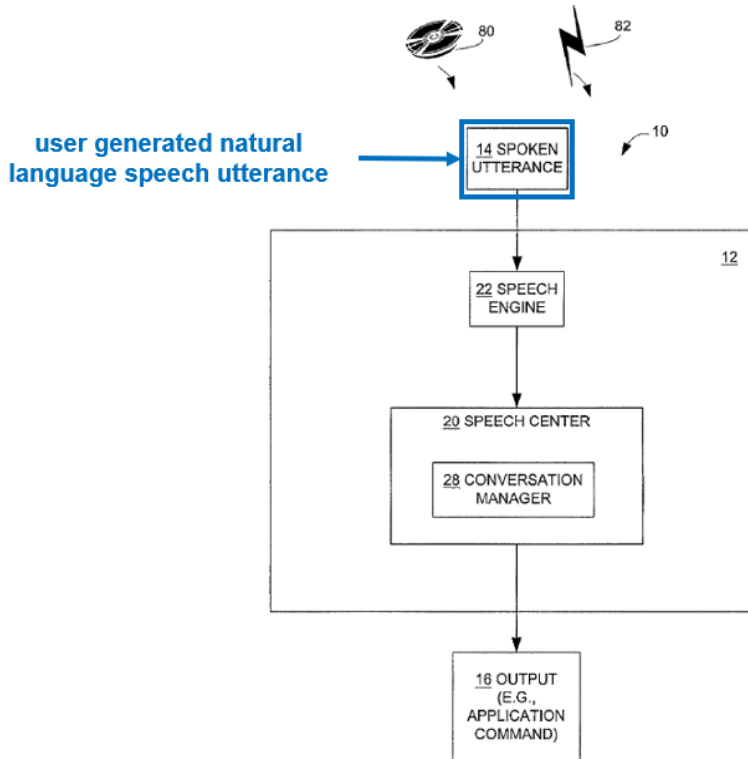


FIG. 1

EX1005, Fig. 1 (annotated)

See also EX1005, Fig. 1, 3:41-53; Lopresti ¶¶137-139.

2. **5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;”**

Ross in view of Ittycheriah renders obvious this limitation. Lopresti ¶¶140-146. As discussed in the preamble, Ross discloses a method that receives a spoken utterance (*user generated natural language speech utterance*) and responds to that spoken utterance using the speech center. EX1005, 5:34-38; Section VII.C.1 (Ground 1, Claim 5[Preamble]). To provide text strings to the speech center, Ross discloses a “speech engine” to recognize words in the spoken utterance as shown in Figure 1:

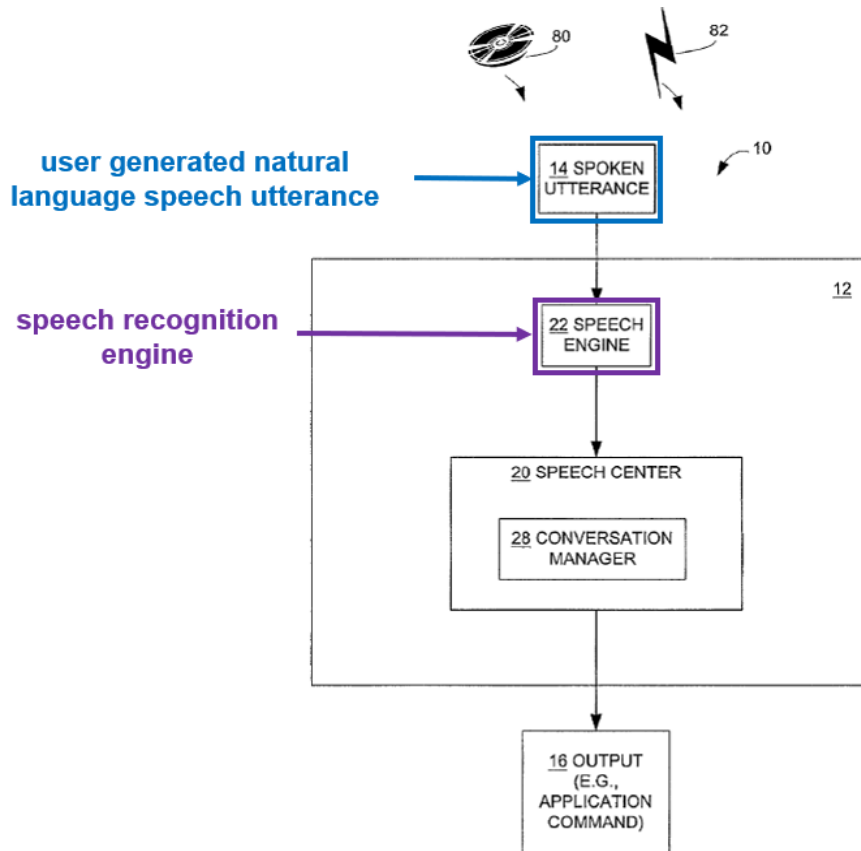


FIG. 1

EX1005, Fig. 1 (annotated)

Ross does not disclose the details of the speech engine, instead teaching that any well-known “speech recognition products” could “convert speech into text strings,” such as the ViaVoice™ from IBM® and NaturallySpeaking Professional from Dragon Systems. EX1005, 1:17-22; Lopresti ¶¶141-143.

Ittycheriah teaches such a speech recognition product also developed by IBM®. EX1007, 1:10-15, [73]. Ittycheriah recognizes one or more keywords (*recognizing, by a speech engine, one or more words*) in a speech segment, such as

the spoken utterance of Ross (*in the user generated natural language speech utterance*) and outputs a “recognized script (name, command or sentence).” EX1007, 3:2-6; Lopresti ¶¶144-145.

A POSITA would have used Ittycheriah’s speech recognition to implement the details of the speech engine in Ross given that both systems were developed by IBM and Ross expressly taught a POSITA to use available speech recognition products as the speech engine. Lopresti ¶146; *see also* Sections VII.A (Motivation to Combine) and VII.B (Reasonable Expectation of Success).

3. **5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;”**

Ross in view of Ittycheriah renders obvious this limitation. Lopresti ¶¶147-163. The ’825 Patent explains that a *parser* “transforms [recognized words and phrases] into complete commands and questions.” EX1001, 11:19-21. Ross’s “speech center” includes a *parser* as part of its “context manager,” because it transforms the received keywords into something that has meaning. Lopresti ¶148; *see also* Section VII.C.4 (Ground 1, 5[c]) and VII.C.5 (Ground 1, 5[d]).

The ’825 Patent explains that a *system agent* “provides default functionality and basic services.” EX1001, 11:57-58. Ittycheriah’s speech recognition is a *system agent* because it is a component of the Ross system that provides the basic service

of recognizing words in the speech utterance and outputs a decoded keyword (*keyword*) along with its likelihood score (*associated prior probabilities or fuzzy possibilities*). Lopresti ¶149.

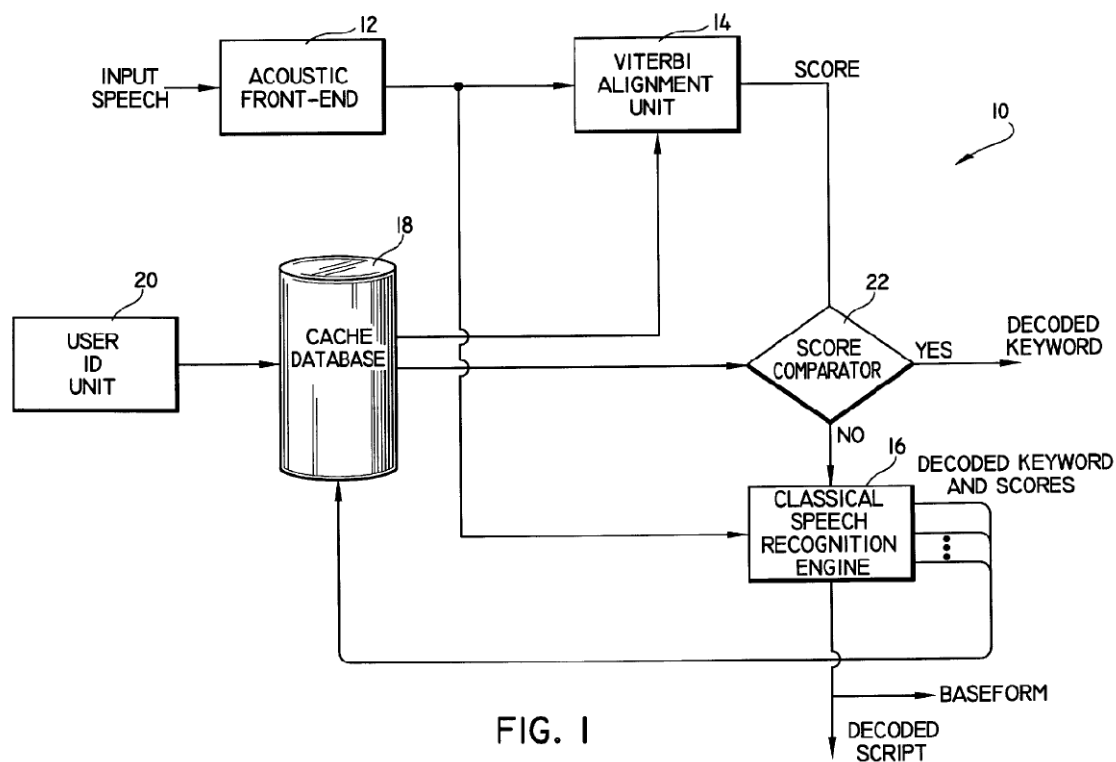
Ittycheriah discloses “storing data in a cache database,” in which the cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” EX1007, 2:63-66. Ittycheriah explains the “Viterbi alignment unit” creates the “likelihood score” using a “probabilistic pattern matching technique using Hidden Markov Models (HMMs)” which “provid[es] different scores for different hypotheses.” EX1007, 3:54-60. Ittycheriah provides different examples of how those “likelihood scores” are defined using *prior probabilities*. EX1007, 3:64-4:4; Lopresti ¶¶151-152.

By June 2002, HMMs were well-known and ubiquitous in the field of speech recognition. *See* EX1019, 15; Lopresti ¶153. HMMs use probability distributions to determine the maximum likely sequence of transitions between a starting state of an input and the probability of a given output. *See id.*, 17-25 (describing how transitions sequences based on probability distributions are calculated by HMMs); Lopresti ¶154. Therefore, Ittycheriah’s cache database that stores “likelihood

scores” are *associated prior probabilities or fuzzy possibilities* of decoded keywords.

Lopresti ¶155.

Ittycheriah describes its cache database as being created from and constantly updated by way of the classical speech recognition engine, labeled as item 16 in Figure 1:



EX1007, Fig. 1; *id.* at 2:66-3:18, 4:33-59; *see id.* at 1:59-2:10. As Ittycheriah explains, once the method receives the input speech (corresponding to the “received utterance”), it applies a probabilistic pattern matching process via the Viterbi

alignment unit 14 to assign various likelihood scores to input speech. EX1007, 4:33-39; Lopresti ¶¶156-157.

The resulting scores are then “presented to the score comparator 22.” EX1007, 4:38-40. The “score comparator” then compares the new scores relating to input speech to the scores of decoded keywords that “were previously stored in the cache database.” EX1007, 4:40-43; Lopresti ¶¶158-161. “If the new scores are not within a margin, or are less than any of the stored scores [in the cache database], then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine 16 whereby a decoded script is generated.” EX1007, 4:48-53.

In the combination, Ittycheriah’s “decoded keyword” corresponds to the “recognition message” in Ross, which is output by the speech engine because it is a “speech representation recognized by the speech engine 22.” EX1005, 7:1-2; EX1007, 4:63-53 (“It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output of the engine 16 (decoded script) may be connected to [external applications] such that the [external applications] are responsive to the decoded keyword or script and thus perform their respective automatic functions.”). A POSITA would have found it obvious to use Ittycheriah’s “decoded keyword” (*keyword and associated prior probabilities or fuzzy*

possibilities) as a recognition message in Ross, which is *received* by Ross’s context manager (*parser*). Lopresti ¶¶162-163; *see also* Section VII.C.4 (Ground 1, 5[c]) and VII.C.5 (Ground 1, 5[d]).

4. **5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;”**

Ross in view of Ittycheriah renders obvious this limitation. Lopresti ¶¶163-170. As discussed in 5[a], Ittycheriah acts as the speech engine in Ross that is used to recognize words and phrases contained in the received utterance and outputs a “recognition message” to Ross’s context manager. EX1005, 6:12-15, 7:12; *see also* Section VII.C.2 (Ground 1, 5[a]). Based on that recognition message (*the received keyword and associated prior probabilities*), Ross discloses *determining...a score for each of at least two possible contexts*. Lopresti ¶166.

Ross *scores* those possible contexts based on the received keyword using the recency of access characteristic (a *relevance score*). Lopresti ¶¶167-170. For example, Ross states that the priority order of context grammars in its context list (such as context list 62) is maintained based on the recency of access characteristic of the context grammars in a context list. EX1005, 6:7-9 (“The context manager 50 maintains the priority and state of the various grammars in the context list 62 in the

system.”); EX1005, 7:17-21; Figs. 4, 5. Ross’s “context manager” then “tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.” EX1005, 8:41-46. Thus, based on the received speech recognition message (the *keyword and associated prior probabilities and fuzzy possibilities*) Ross *determin[es]...scores* for the possible contexts within its context list using recency of access such that “the system 20 is biased to send the commands to the most recently accessed application 26.” EX1005, 8:54-9:6.

5. **5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;”**

Ross discloses this limitation. Lopresti ¶¶171-176.

Ross describes how its context manager (*parser*) *determines a domain for the user generated natural language utterance based on the recognized one or more words and the determined score*. Lopresti ¶¶172-174. Ross gives an example based on multiple ambiguous “print it” or “print this” requests to mail and calendar applications:

1. The user opens the first message in the electronic mail application 26, which results in the context 70 for the electronic mail application

- 26 being placed first in the context list 62. If the user says “print it,” then the first message is printed.
2. The user then opens the second appointment in the calendar application 26, which results in the context 70 for the calendar application 26 being placed first in the context list 62. If the user says “print this,” then the second appointment is printed.
 3. The user next opens the third message in the electronic mail application 26, which results in the context 70 for the electronic mail application 26 being placed first in the context list 62.
 4. The user then uses the mouse to click on the fourth appointment displayed in a window for the calendar application 26. As a result, the context 70 for the calendar application 26 supersedes the context 70 for the electronic mail application 26 in the context list 62. If the user says “print it,” then the fourth appointment is printed.

EX1005, 8:54-9:6. Accordingly, Ross has identified *at least two possible contexts*—a calendar application and an electronic mail application. EX1005, 8:54-9:6.

Ross’s context manager *determines a domain for the user generated natural language utterance* (either the calendar application or electronic mail) based upon *the recognized word* “print it” or “print this” and based on the relevancy of access characteristic (*the determined score*). EX1005, 9:7-20; Lopresti ¶¶175-176. As Ross explains, “the context manager 50 resolves the ambiguous phrase (e.g., “print this” or “print it”) by looking in the context list 62 to determine the top priority

context 70. Then assuming the grammar for the top priority context 70 can process (i.e., recognize) the phrase, the context manager 50 directs a translation of the phrase to the application 26 indicated by the top priority context 70.” EX1005, 9:7-20, Fig. 5.

6. **5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;”**

Ross discloses this limitation. Lopresti ¶¶177-194.

Ross discloses *one of the plurality of autonomous executable domain agents*. Lopresti ¶¶178-180. The '825 Patent describes “domain agents” as “autonomous executables that receive process, and respond to user questions, queries, and commands,” that “provide complete, convenient and redistributable packages or modules of functionality.” EX1001, 2:50-59, 2:41-42. A POSITA would thus understand *domain agent* includes autonomous executable software modules specific to each application area. Lopresti ¶180.

Ross discloses a *plurality* of “external applications 26” (*domain agents*) that “communicate with the speech center 20, a speech engine 22, and active accessibility

module **24**.” EX1005, 4:16-21. Each external application has a defined grammar, “along with associated translations for a script engine **38**.” EX1005, 6:29-31. Further, each external application *is configured to respond to queries and/or commands within a particular domain*, as each is a “speech-enabled application” that respond to speech. EX1001, Abstract, 2:22-36; Lopresti ¶¶181-183. As Ross explains:

The speech center can allow cross-application speech. In one example, a user is working with a spreadsheet application, but then asks about the user’s next appointment and obtains the information from a calendar program. In another example, the user asks what is the current stock price for a particular company and obtains the stock price from a stock monitoring service. Furthermore, the user asks whether the user has any new messages and receives an answer supplied by a message or mail application.

Applications do not even have to be running (i.e., currently performing on a computer system). For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.

EX1005, 2:22-36.

Ross discloses a task manager that “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of execution.” EX1005, 5:1-5.

Based on these disclosures, a POSITA would have thus understood that the external applications in Ross are autonomous executables triggered by the script commands issued by Ross's task manager. Lopresti ¶¶186. Accordingly, each of Ross's external, speech-enabled applications is a *domain agent* as claimed in the '825 patent. Lopresti ¶¶39-45, 187; EX1012. This also aligns with PO's infringement contentions, in which Dialect only identifies external applications that can be called as "domain agents." EX1015, 12-13.

Ross further discloses *selecting* the domain agent *at least in part, based on the determined domain*. Lopresti ¶¶188-190. Specifically, Ross "maintains a context list of speech enabled applications" that the user has accessed and teaches "deciding the target application (*selecting a domain agent*) for a received utterance based in part on the determined context (*determined domain*). Ross teaches that the "associated application 26 is the one that should be targeted to receive the corresponding command" if that syntactic parse is accepted by the grammar of a specific speech-enabled external application. EX1005, 8:33-36.

Ross describes how *the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands*. Lopresti ¶¶191-194. For example, Ross describes how requests regarding appointments and dates (*the particular domain*) is an area of expertise for

a calendar application (*the domain agent*). EX1005, 2:22-25. Ross also describe how stocks (*the particular domain*) is an area of expertise for a stock monitoring service (*the domain agent*). EX1005, 2:25-28, 2:28-30, 2:32-36. Accordingly, a POSITA would understand that Ross discloses this limitation. Lopresti ¶194.

7. **5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;”**

Ross discloses this limitation. Lopresti ¶¶195-201. Ross teaches that the natural language utterance contains *at least one query and/or command*. Lopresti ¶¶196-197. Ross discloses that “[t]he input spoken utterance 14 is a voice command or other speech input from a user of the computer system 10 (e.g., into a microphone connected to the computer system 10). EX1005, 3:45-48. A POSITA would understand that a “voice command” is *at least one query and/or command*. Lopresti ¶198; EX1005, 3:45-48, 9:40-43 (“[T]he spoken utterance including a command that is able to be associated with multiple applications, any subset of which may or may not be running[.]”), 2:22-36.

Ross further discloses *providing the at least one query and/or command to the selected at least one of the plurality of domain agents*. Lopresti ¶¶199-201. After selecting the target external application (i.e., *domain agent*) as discussed in 5[e], Ross translates the received utterance to be processed by the external application and

calls and dispatches the request to the target external application (*at least one of the plurality of domain agents*). See EX1005, 5:55-57 (“The utterance is then translated into the appropriate script engine 38 calls and dispatched to the target application 26), 6:15-18 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution[.]”), 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26), 7:26-29 (“In step 108, the context manager 50 direct the speech representation to be translated according to the selected context 27[.]”), 9:9-13, Abstract, Fig. 5 (step 108). This translation is associated with the “grammar defined for each application.” EX1005, 6:29-31 (“Toward that end, a grammar is defined for each application 26, along with the associated translations for a script engine.”).

8. **5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”**

Ross discloses this limitation. Lopresti ¶¶202-205. Ross’s external applications (*one of the plurality of domain agents*) create...*one or more queries based on the at least one query and/or command*. Lopresti ¶¶203-204. Once the target external application (*domain agent*) is selected, Ross translates the received utterance to be processed by the external application. EX1005, 5:55-57 (“The

utterance is then translated into the appropriate script engine 38 calls and dispatched to the target application 26), 6:15-18 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution[.]”), 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26), 7:26-29 (“In step 108, the context manager 50 direct the speech representation to be translated according to the selected context 27[.]”), 9:9-13, Abstract, Fig. 5 (step 108). This translation is associated with the “grammar defined for each application.” EX1005, 6:29-31 (“Toward that end, a grammar is defined for each application 26, along with the associated translations for a script engine.”).

A POSITA would have understood Ross’s “translation” of the received utterance would create a query based on the at least one natural language query or command. Lopresti ¶205; *see also* EX1005, 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26). This also aligns with PO’s infringement contentions, which identifies an external application querying information sources based on the natural language input as satisfying this limitation. EX1015 at 13-14.

9. **5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”**

Ross discloses this limitation. Lopresti ¶¶206-214. A POSITA would understand that the term “asynchronous” in the programming context includes sending the queries to be executed autonomously, rather than being controlled by the main programming flow. Lopresti ¶¶46-47, 207-208. This is a well-known paradigm of computer programming that allows for different tasks to be performed independently of the other. Lopresti ¶209. This aligns with how the ’825 Patent describes sending “asynchronous” queries to “account for the fact that sources respond at different speeds or may fail to respond at all.” EX1001, 19:57-61, 24:13-17 (“In addition, the use of asynchronous queries that may result in rapid and graceful failure of some queries or commands may allow the system 90 to robustly return results quickly and in a manner that seems natural to the user.”).

Ross discloses that its external applications (*domain agents*) operate autonomously and thus send requests and/or commands *asynchronously* to *local or external information sources* apart from the main programming flow of recognizing and processing the speech input. Lopresti ¶¶210-211; EX1005, 9:46-50 (“[T]he command is applied to the speech enabled application as needed and performing the function of the command.”), 10:19-23 (“[T]he command is applied to the selected

speech enabled application as needed and performing the function of the command.”), 12:6-13. Moreover, Ross discloses that its method has “the capability to proceed with multiple execution requests simultaneously,” demonstrating asynchronous programming. EX1005, 5:1-7.

As discussed in 5[g], the external applications (*by the selected at least one or more domain agents*) create a “translated” query based on the at least one natural language query or command (*one or more queries based on the at least one query and/or command*). See Section VII.C.8 (Ground 1, 5[g]). A POSITA would have further understood that the external applications (*domain agents*) send the command or query to the target application (*a local or external information source to execute the request*) as part of executing the function of the command. EX1005, 5:1-7; Lopresti ¶¶212-213. Ross explains that after the utterance is translated into the appropriate commands or queries, they are then “dispatched to the target application 26.” EX1005, 5:55-56. For example, for electronic mail application 26, commands may include “Open the first message” or “Create a message.” EX1005, 8:1-6. Similarly, for calendar application 26, commands may include “Open the first appointment” and “Create an appointment.” EX1005, 8:19-24. The external applications (*domain agents*), would send these commands to the electronic mail or calendar sources (*local or external information sources*) to execute. Lopresti ¶214.

D. Dependent Claims 6-7

1. **Claim 6: “further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.”**

Ross discloses this limitation. Lopresti ¶¶215-216. Ross discloses that the context manager generates (*generating, by the parser*) the translated utterance (*at least one query and/or command*) based on the spoken language utterance (*natural language utterance*). EX1005, 7:29-33 (“In step 110, the context manager 50 sends the translated utterance 74 to the speech enabled application 26 indicated by the selected context 72 to perform the action indicated by the translated utterance 74.”), Figs. 4-5; Lopresti ¶216.

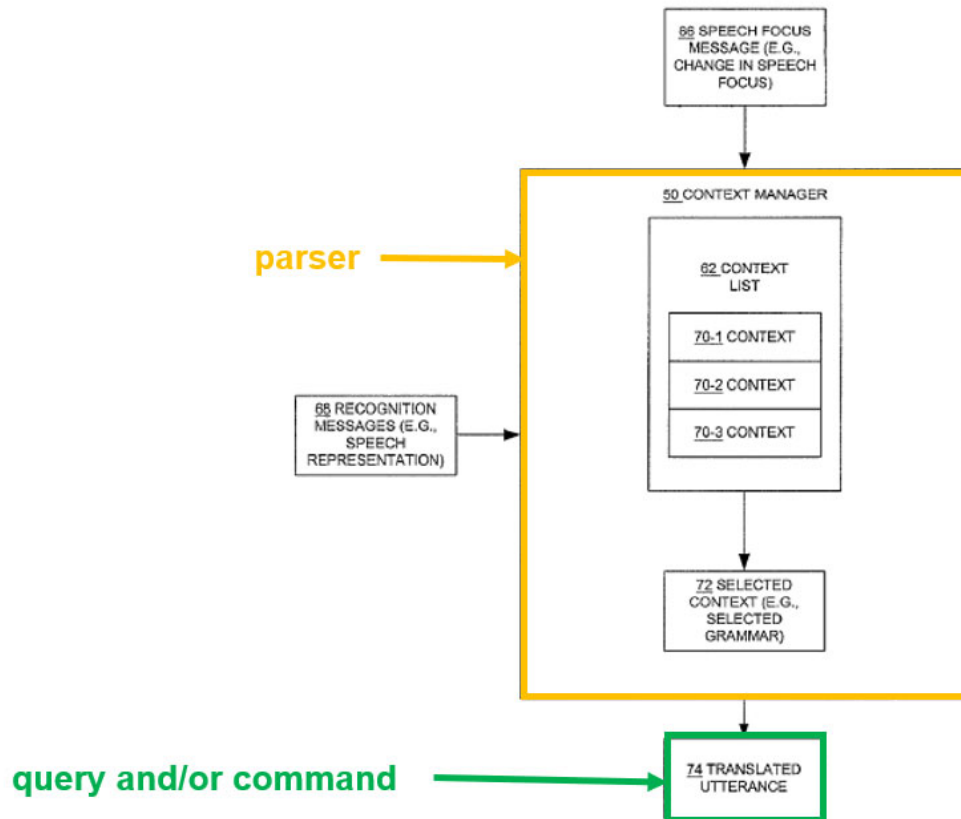


FIG. 4

EX1005, Fig. 4 (annotated)

2. **Claim 7: “wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”**

Ross discloses this limitation. Lopresti ¶¶217-219. As discussed for claim 6, Ross discloses that the context manager generates (*generating, by the parser*) the translated utterance based on the spoken language utterance (*at least one query and/or command based on the natural language utterance*). See Section VII.D.1 (Ground 1, Claim 6). Ross further discloses that the translated utterance (*at least*

one query and/or command based on the natural language utterance) can include commands to multiple applications (*comprises a plurality of queries and/or commands*). Lopresti ¶¶218-219.

Ross explains that the speech center “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of the execution, informing the client when execution of a script is in progress or has completed.” EX1005, 5:2-7. Accordingly, “[t]he spoken utterance [can] include[e] a command that is able to be associated with multiple applications.” EX1005, cl. 1. This allows the system to avoid “the time, expense, and footprint of duplicating speech services in multiple applications.” EX1005, 2:14-16.

VIII. GROUND 2

A. Motivation to Combine

A POSITA would have been motivated to implement Glover’s metasearch functionality in Ross’s system, such that the external applications (*domain agents*) can gather relevant information from the Internet. Lopresti ¶¶221-226. Ross discloses that its external applications “perform the function of the command” in the natural language utterance. EX1005, 9:44-50, 12:6-13. Ross does not detail how all its external applications obtain and process information to perform the function of the command (including sourcing information from the Internet). Accordingly, a

POSITA would have been motivated to look for and rely on an Internet searching mechanism to implement retrieving and processing responses to a user's queries (like in Glover). Lopresti ¶222.

A POSITA would have looked to implement a metasearch engine like Glover that includes a dispatcher that operates "similar to the query processor of a conventional search engine," as it "generate[s] search engine requests from the user's input," to multiple search engines. Lopresti ¶223; EX1011, ¶38. A POSITA would have found a metasearch engine, like Glover, that interfaces with Ross's system for a broad search over several Internet databases desirable as it provides "improve[d] coverage compared to a single search engine." Lopresti ¶223; EX1011, ¶35.

A POSITA would have been motivated to implement Glover's metasearch functionality because it uses "selective retrieval," which "predicts the relevance of documents returned by regular search engines." Lopresti ¶224; EX1011, ¶5. Specifically, Glover discloses a metasearch engine that performs continuous searching based on whether the returned results exceed a pre-determined confidence threshold. EX1011, ¶83. For those documents below the threshold, Glover's metasearch engine performs additional searches until it has confidence that the returned result provides relevant results. EX1011, ¶83. This "computation of the

confidence of the relevance prediction,” can be done using “similarity measures..., or machine learning methods such as neural networks or support vector machines.” EX1011, ¶¶85-86.

By adopting Glover’s “selective retrieval,” Ross’s ability to search the Internet and return results would benefit from “improve[d] execution time, resource usage, and throughput by requiring fewer retrieval requests compared to content-based metasearch engines.” EX1011, ¶5; Lopresti ¶225. Thus, Glover advances Ross’s objective of “avoid[ing] the time, expense, and footprint of duplicating speech services in multiple applications.” EX1005, 2:14-16.

Accordingly, a POSITA would have found Glover particularly well-suited to include in a speech-based system because it automates many steps that a user would otherwise have to perform manually in refining a search to obtain good results. Lopresti ¶226. Those search refinements would otherwise be more difficult to achieve with a speech interface, so a POSITA would have found implementing Glover’s metasearch engine in Ross an improvement. *Id.*

B. Reasonable Expectation of Success

A POSITA would have easily been able to implement a metasearch engine as Ross’s external applications and to allow for Ross’s system to invoke Internet searching. Lopresti ¶¶227-230.

Glover's metasearch engine simply extends Ross's ability to process natural language commands and questions by expanding the sources of information it can search to information on the Internet. Lopresti ¶228.

Ross teaches that its external applications can communicate with Ross's conversation manager and speech center via the "external application interface." EX1005, 5:8-21. Because Ross already provides an interface for the external applications (*domain agents*) to communicate with Ross's conversation manager, it would be a simple adjustment to implement a network interface to Glover's dispatcher to obtain information from the Internet to respond to a user's query or command. Lopresti ¶229.

A POSITA would have had the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and Internet searching based on user queries) according to known methods (speech recognition engines that convert speech into text strings and using text strings representing user's queries to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications to retrieve information from the Internet. Lopresti ¶230.

C. Independent Claim 5

1. Preamble through 5[f]

Ross in view of Ittycheriah renders obvious the preamble and limitations 5[a]-5[f] for the reasons discussed in Section VII.C.1-7 (Ground 1, Claim 5[preamble]-5[f]). Lopresti ¶232.

2. 5[g]: “creating...”

To the extent Ross alone does not disclose 5[g], Ross in view of Glover renders obvious this limitation. Lopresti ¶¶233-235. Specifically, Glover’s dispatcher creates (*creating, by the selected at least one of the plurality of domain agents*) database queries based on the Ross’s translated utterance (*one or more queries based on the at least one query and/or command*). EX1011, ¶38 (metasearch engine includes a dispatcher that “generate[s] database queries based on the input from the user interface.”), ¶39 (dispatcher includes a “query generator 32,” that “modif[ies] queries appropriately for each source.”); Section VIII.A (Ground 2, Motivation to Combine).

3. 5[h]: “sending...”

To the extent Ross alone does not disclose 5[h], Ross in view of Glover renders obvious this limitation. Lopresti ¶¶236-239. Specifically, Glover’s dispatcher sends (*sending, by the selected at least one of the plurality of domain agents*) the database queries (*the one or more queries*) to multiple search engines on

the World Wide Web (*to one or more local or external information sources*).
EX1011, ¶35 (“A metasearch engine takes user queries and submits them to multiple underlying search engines, and combines the results into a single interface.”), ¶39, Fig. 3.

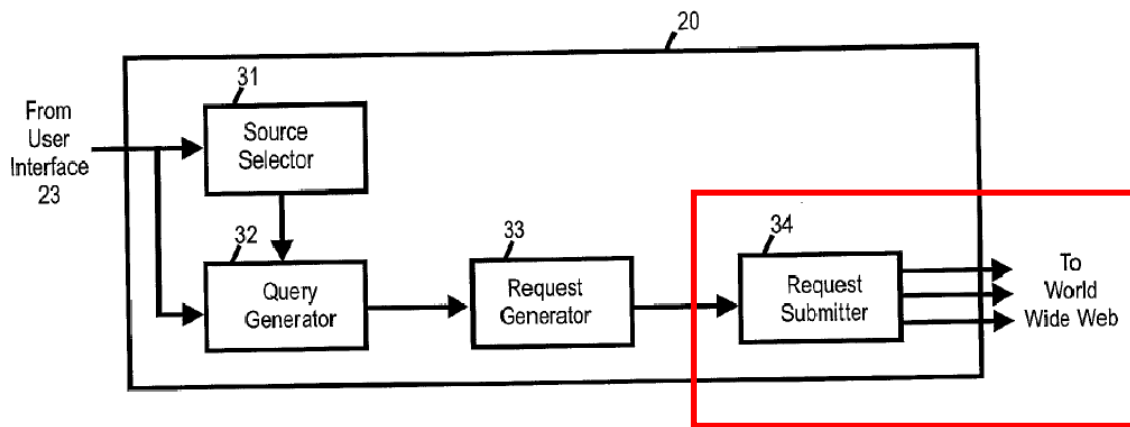


FIG. 3

EX1011, Fig. 3 (annotated)

A POSITA would have further understood that a query submitted to “multiple underlying search engines” is sending a query to “one or more local or external information sources.” Lopresti ¶238.

A POSITA would have also understood that Glover sends the queries *in an asynchronous manner*, as Glover states that its “steps may be performed in parallel.” Lopresti ¶239; EX1011, ¶82; *see also id.* (“For example, step 53 may be estimating the relevance of one or more results while step 52 is still sending queries 51 to and retrieving results 52 from search engines.”).

D. Dependent Claims 6-8

1. **Claim 6: “further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.”**

Ross discloses this limitation for the reasons discussed in Section VII.D.1 (Ground 1, Claim 6). Lopresti ¶240.

2. **Claim 7: “wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”**

Ross discloses this limitation for the reasons discussed in Section VII.D.2 (Ground 1, Claim 7). Lopresti ¶241.

3. **Claim 8: “wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.”**

Ross in view Glover renders obvious this limitation. Lopresti ¶¶243-246. As discussed for 5[g] and 5[h], in the combination, Ross’s conversation manager mediates between the external applications (*domain agents*) including Glover’s metasearch engine such that the external applications create and send queries to different information sources based on the user’s input. Lopresti ¶244. Specifically, Glover’s dispatcher (*domain agent*) will send queries to various search engines (*different information sources*). *Id.*

Glover discloses that generally, “a metasearch engine takes user queries and submits them to multiple underlying search engines.” EX1011, ¶35. Glover further discloses that its dispatcher can “choose search engines to query and...modify queries appropriately for each source.” *Id.*, ¶39. A POSITA would have understood that taking a single user query and submitting to multiple search engines discloses *creating duplicate queries and sending the duplicate queries to different information sources*, as a query must be duplicated to be sent to multiple sources. Lopresti ¶245; Fig. 3 (showing “request submitter” sending duplicate queries to the World Wide Web).

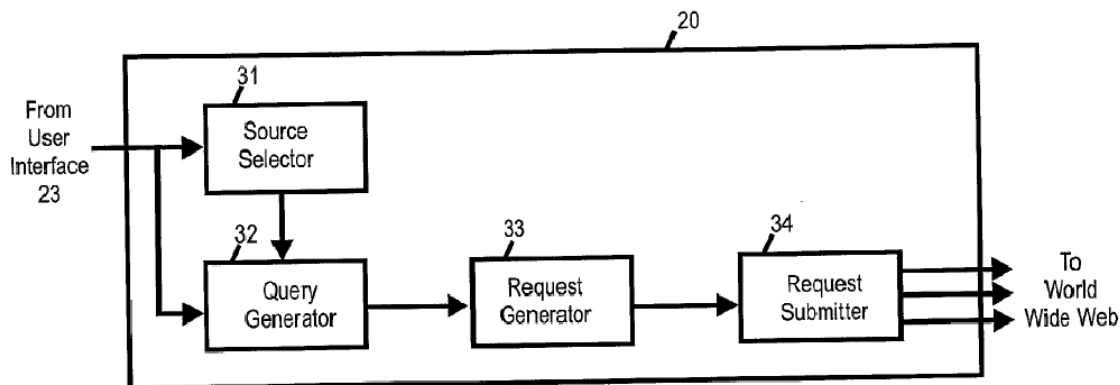


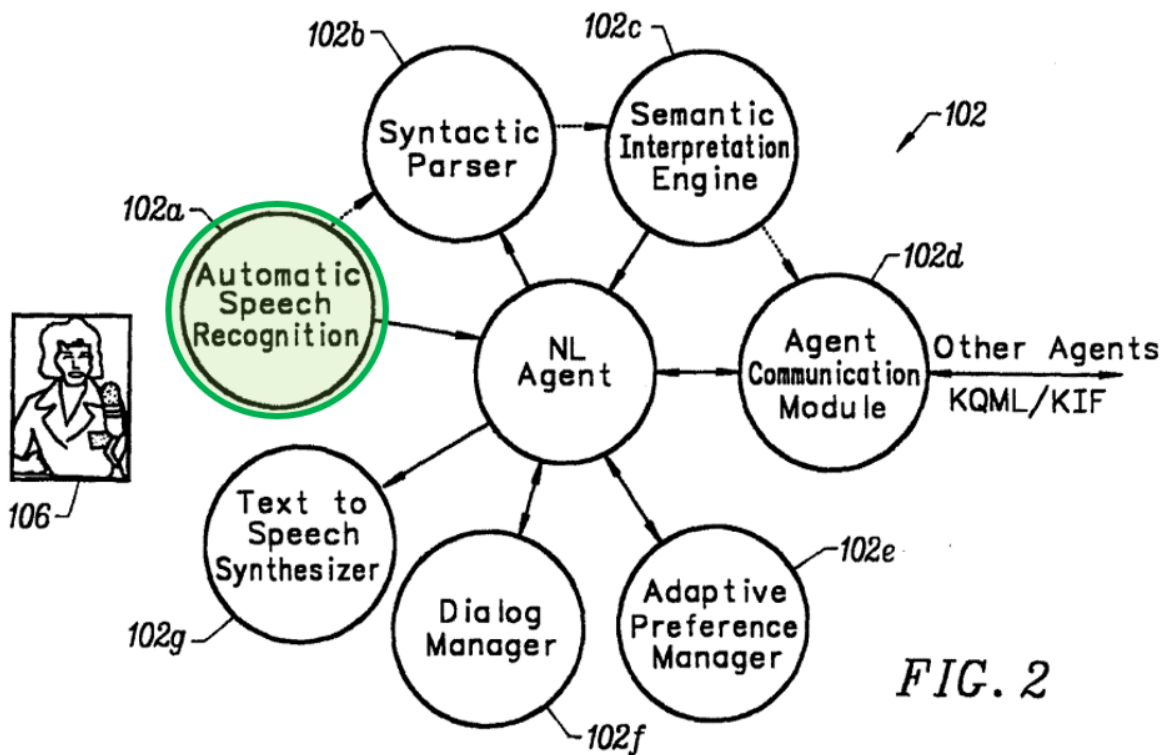
FIG. 3

For the same reasons as stated for 5[h], a POSITA would also understand that Glover discloses sending the duplicate queries *in an asynchronous manner*. Lopresti ¶246; Section VIII.C.3 (Ground 2, 5[h]); EX1011, ¶82.

IX. GROUND 3

A. Motivation to Combine

A POSITA would have been motivated to use Baker's speech recognition method as Hartono's "automatic speech recognition system," to implement a way to quickly complete the speech recognition process based on both acoustic probabilities and context probabilities. Lopresti ¶¶248-256. Hartono discloses that speech is input into an "automatic speech recognition" module, shown below labeled as item 102a in Figure 2:



Hartono recognizes that "automatic speech recognition systems for speech input are readily and commercially available off the shelf." EX1006, 11:4-12. Hartono thus

teaches that “[a]ny suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a in the natural language interface system 100 of the present invention.” EX1006, 11:4-12. A POSITA reading Hartono would thus be motivated to look for commercially available speech recognition systems to use to perform the method steps described in Hartono. Lopresti ¶¶248-251. Baker is such a speech recognition system developed by Dragon Systems, Inc., a well-known developer of speech recognition software in the mid-to-late 1990s. EX1008, [73]; Lopresti ¶252.

A POSITA would have also been motivated to implement the speech recognition system of Baker to quickly complete the speech recognition process based on both acoustic probabilities and context probabilities by providing a probability and rank signal that Hartono can then use to determine which context is most suitable. Lopresti ¶253.

A POSITA would have also been motivated to use Baker’s speech recognition system with Hartono’s because Baker recognizes words “on the fly” using existing and on-line reference sources. Lopresti ¶¶254-255. This allows Baker to process commands quickly over those methods that require building data-intensive statistical language models or data-training sets. EX1008, 3:46-4:56 (describing the drawbacks of prior art systems based on “precomputed statistical counts” and

“vocabulary training.”), 4:57-61, 6:18-22, 13:66-14:13. A POSITA would be motivated to use Baker’s faster processing in speech recognition to more quickly respond to specific user requests. Lopresti ¶256.

B. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in combining the teachings of Hartono and Baker. Lopresti ¶¶257-260.

A POSITA would have understood that Baker’s speech recognition system would be compatible with Hartono, as Baker discloses recognizing “words, or a series of words,” for “translat[ing] into machine readable form.” EX1008, 1:5-15; Lopresti ¶258; *see also* EX1006, 11:4-12.

Hartono also employs the Government and Binding approach to its syntactic parsing, which is based on the “implied syntactic structure in English language sentences.” EX1006, 12:1-10. Baker’s speech recognition system that recognizes words based on their context within sentences, is thus compatible with Hartono’s syntactic parser, which looks aims to accurately identify noun phrases, verb phrases, and adjective phrases. Lopresti ¶259; EX1006, 12:15-13:10.

A POSITA would have also had the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (e.g., known systems for speech

recognition and processing natural language requests) according to known methods (e.g., known methods of speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications on a computer. Lopresti ¶¶260.

C. Independent Claim 5

1. Preamble

To the extent limiting, Hartono discloses the preamble. Lopresti ¶¶262-264. Hartono discloses a “method for receiving commands and/or adaptively outputting results using a natural language speech interface[.]” EX1006, 8:8-10; *see also* 1:19-23 (“provides a method for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.”), 4:20-25 (“In one embodiment, the natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input and to output an output command and at least one application agent adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 5:16-24, 10:13-20.

2. 5[a]: “recognizing...”

Hartono in view of Baker renders obvious this limitation. Lopresti ¶¶262-

264.

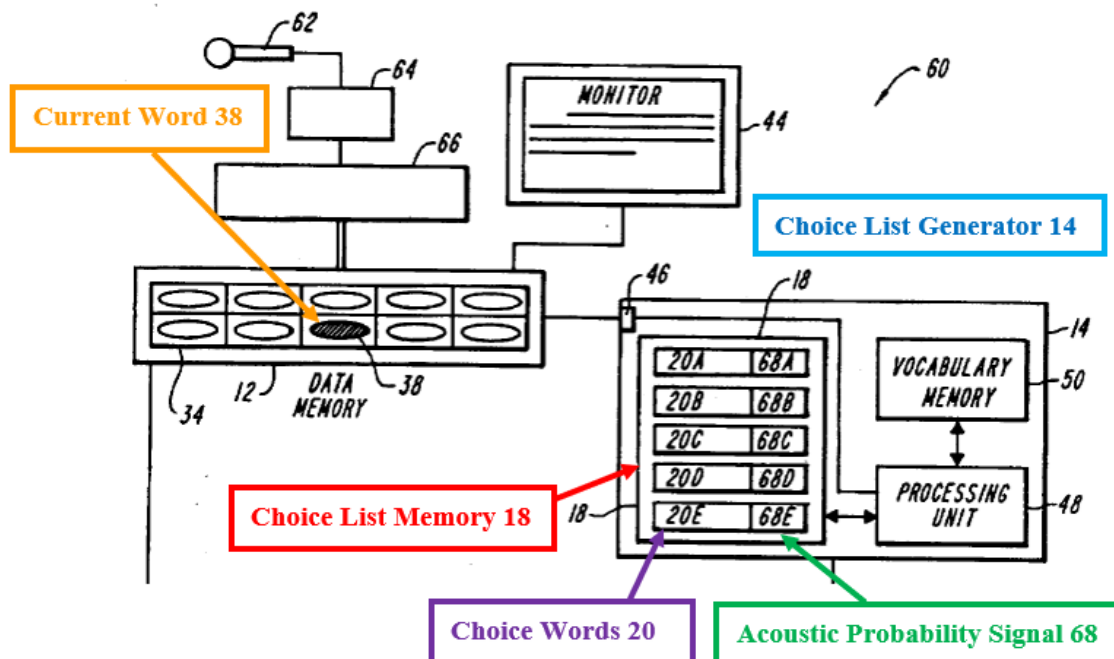
Hartono, recognizes that “[a]utomatic speech recognition systems for speech input are readily and commercially available off the shelf. Any suitable off the shelf speech recognition systems may be used as the automatic speech recognition system **102a** in the natural language interface system **100** of the present invention.” EX1006, 11:4-12. Thus, a POSITA would have been motivated to combine Hartono with an off-the-shelf speech recognition system, like that disclosed in Baker. Lopresti ¶¶266-268.

As discussed in Section IV.D (Baker Overview), Baker’s word recognition system is a *speech recognition engine* that *recognizes one or more words in the user generated natural language speech utterance*. Baker teaches recognizing words in two ways. Lopresti ¶¶270-280. First, Baker describes systems and methods for “detect[ing] words within portions of speech signals and compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” EX1008, 18:13-16.² “This comparison leads to a probability signal that

² Notably, Baker defines the term “word” to mean “a word or a *phrase* and may be spoken, written, or represented as electronic data signals.” EX1008, 6:1-3 (emphasis added); Lopresti ¶¶271-272.

indicates the certainty that the detected word is substantially acoustically similar to the word associated with the acoustic model.” EX1008, 18:16-19. Words with an acoustic probability signal indicating a “high degree of certainty” are deemed recognized (referenced as “context words”). EX1008, 18:16-19. In contrast, words with a “low degree of certainty” indicate that the word has not been confidently identified (referenced as “current word”). EX1008, 18:22-24.

Second, for those “unrecognized words,” Baker creates a “choice list **18**” that “store[s] the words **20** associated with the acoustic models from the vocabulary memory **50** which are acoustically similar to the current ambiguous word **38** being identified, as illustrated in Figure 2:



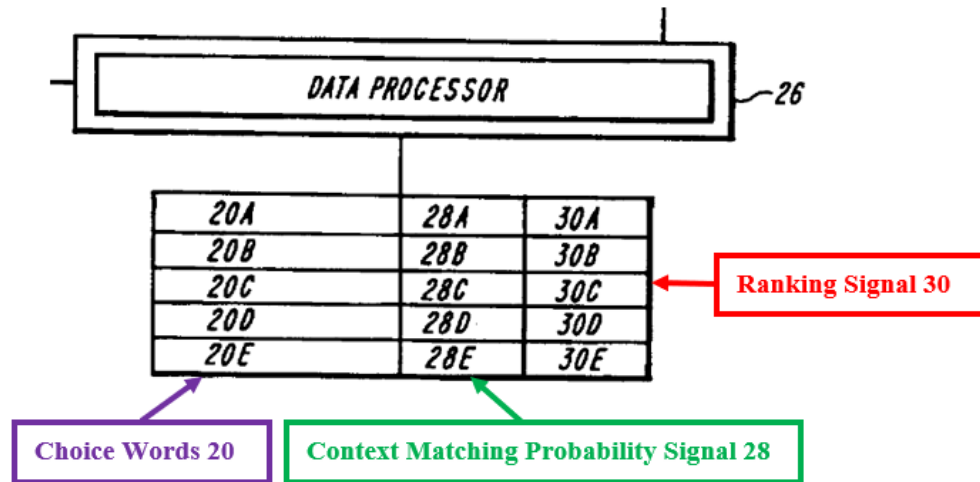
EX1008, Fig. 2 (annotated)

Lopresti ¶276.

Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represent “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” EX1008, 18:30-35. Baker further teaches that probability signals 68A-E “can represent the acoustic model probability adjusted by an n-gram probability, or any suitable language model.” EX1008, 18:35-37, 20:30-46 (“The probability signals 68A-68E can represent the likelihood that the associated word choice 20A-20E is substantially representative of the current word 38 being recognized.”), 22:55-64. Once a “choice word 20” has been identified, Baker teaches generating a “word list” of words that “have a semantic association with the choice word 20.” EX1008, 14:57-69. The “semantic association” for each choice word (which Baker also refers to as “passage words” (labeled as item 24 in Fig. 2)) in the “word list” is derived from a “reference database 40” which can be a “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” EX1008, 20:55-58 (“The word list generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that

relate to the topic, theme, or subject of a choice word 20.”); Lopresti ¶¶277-280.

Baker next “perform[s] a matching function to calculate the number of matched words” between those words recognized by the speech analysis unit 66 with high confidence (the “context words”) with the words in the “word list” (the “passage words”). EX1008, 21:19-47. By looking at the “relative frequency of matches between the “context words” and the “passage words,” the data processor calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” EX1008, 21:19-47, 21:49-22:6, Figs, 2-4. Baker generates a “rank signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized” as a function of “context matching and acoustic model matching” probabilities. EX1008, 23:22-27. A table correlating each “choice word 20” (words selected based on acoustic-matching) with context-matching “probability signal 28” and “rank signal 30” is shown in Figure 2:



EX1008, Fig. 2 (annotated)

Lopresti ¶284; *see also* EX1008, 21:48-22:6 (describing computing rank signal 30 “given two constraints, the probability signal 28 and the probability signal 68” according to techniques “well known in the field of mathematics and any of the techniques capable of generating a probability signal given a set of constraints”); Lopresti ¶¶281-283. By looking at both acoustic matching and context matching, Baker is thus able to *recognize one or more words in the user generated natural language speech utterance*. Lopresti ¶285.

3. 5[b]: “receiving...”

Hartono in view of Baker renders obvious this limitation. Lopresti ¶¶286-291.

Baker’s speech recognition system is a *system agent* because it is a component of Hartono’s system that provides the basic service of recognizing words in the

utterance and outputs words or phrases along with probability and rank signals. Lopresti ¶¶287-290; *see also* Section VII.C.3 (Ground 1, 5[b]) (discussing the '825 Patent's description of *system agent*). In the combination, Baker's output is the recognized words or phrase of a spoken utterance "just as if the data were being typed." EX1008, 1:4-7, 1:32-36. Those recognized words or series of words (*keyword*) along with a probability and rank signal (*associated prior probabilities or fuzzy possibilities*) are then inputted into Hartono's natural language "syntactic parser" and "semantic interpreter." EX1006, 11:14-16:11, Fig. 2; *see also* Section IX.C.4 (Ground 3, 5[c]). Hartono receives the recognized speech from Baker and its associated prior probabilities or fuzzy possibilities at its "natural language syntactic parser 102b" and "natural language semantic interpreter 102c" (*parser*). *See* Section IX.C.4 (Ground 3, 5[c]).

Thus, in the proposed combination, Hartono's syntactic parser and semantic interpreter receives (*receiving, at a parser*) recognized words (*keyword*) and probability and rank signals (*associated prior probabilities or fuzzy possibilities*) from Baker's speech recognition system (*from a system agent*). Lopresti ¶291.

4. **5[c]: "determining...a score..."**

Hartono in view of Baker renders obvious this limitation. Lopresti ¶¶292-308. Hartono's "natural language syntactic parser 102b" and "natural language

semantic interpreter 102c” receive the recognized words from Baker’s speech recognition engine. Hartono then discloses *determin[ing]...a score for each of at least two possible contexts for the natural language speech utterance*. Lopresti ¶¶293-294.

Specifically, Hartono discloses syntactically parsing the recognized word into a structure using a Government and Binding approach (a computation method based on linguistic theory). EX1006, 12:1-20. This parsing “allow[s] the computer to map the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as properties of the object.” EX1006, 13:8-10. Hartono’s semantic interpreter then applies “context sensitive methodologies” to the syntactic parse to *determine a score for each of at least two possible contexts*. EX1006, 13:13-21; Lopresti ¶¶295-297.

Namely, Hartono’s semantic interpreter uses a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” EX1006, 14:10-20. For example, “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%”), resulting in different permutations of the natural language utterance, scored from highest to lowest probability. EX1006, 14:10-20. Using these different probability-based

permutations, Hartono “determines which words best fit the grammar of the syntactic parse,” which are *determined scores for possible contexts*. EX1006, 15:1-12. A POSITA would have understood that to determine “best fit” a value or score would need to be assigned to the evaluated grammar of the syntactic parse to determine whether or not the word is “suitable given the context.” Lopresti ¶¶298-306.

The suitability given the context is based on the recognized word and probability and rank signals (*wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities*). Lopresti ¶307. Specifically, Hartono will iterate through the recognized words received from Baker depending on the probabilities and ranks provided. *See* EX1006, 15:8-12 (“If the word is not suitable given the context, the next word having each next highest probability is then evaluated and determined if that word is suitable given the context, until a suitable word is determined.”).

Thus, Hartono discloses *determining, for the natural language speech utterance, the best fit (a score) for the grammar of the syntactic parse for the different permutations of the words (for each of at least two possible contexts) based on the associated vector or list and their corresponding probabilities (wherein the*

scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities). Lopresti ¶308.

5. **5[d]: “determining...a domain...”**

Hartono discloses this limitation. Lopresti ¶¶309-313. Once the semantic interpreter has scored the words in the recognized natural language utterance, it then *determines a domain for the user generated natural language utterance* by “take[ing] the syntactic parse of the spoken language request and map[ing] it to a generic concept frame,” such as that shown below in Table 1:

TABLE I		
ACTION-CONCEPT	TOPIC CONCEPT	APPLICATION-CONTEXT
Show	Email	Email application
Show	Email address	Address book application
Delete	Email	Email application
Show	General help	Natural language agent

EX1006, 13:18-21, Table I, Table III. Specifically, the semantic interpreter uses verbs in the recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” EX1006, 14:1-7. Hartono “uses the combination of the action-concept and the topic-concepts” to “determine which task agent should handle the request.” EX1006, 15:13-19. The *domains* are shown in Table I as the “Application-Context,” as “email application” or “address book application.” EX1006, Table I, 6:2-5 (“The natural

language interface parses the user's input and semantically maps it into a knowledge concept structure. The system then determines which application context should be responsible for interpreting and executing that command concept.”).

6. **5[e]: “selecting...”**

Hartono discloses this limitation. Lopresti ¶¶314-325. As discussed for Ground 1, a POSITA would thus understand a “domain agent” as claimed to be an autonomous executable software module specific to each application area. *See* Section VII.C.6 (Ground 1, 5[e]).

Like the '825 patent, Hartono “utilizes an agent-based architecture” that uses “application class task agents for each class of applications,” as illustrated in Figure 1:

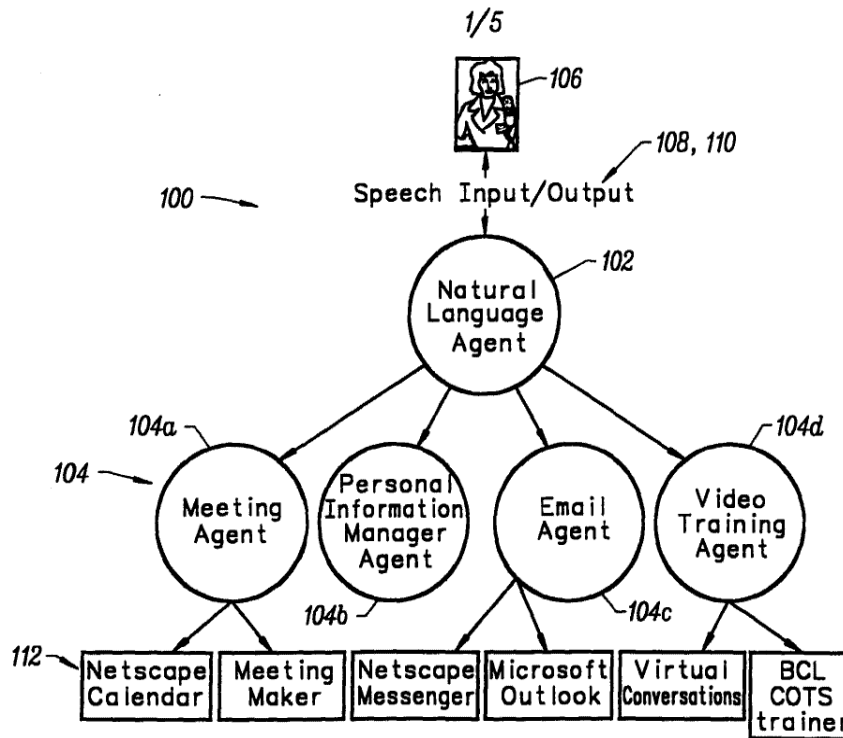


FIG. 1

EX1006, Fig. 1

EX1006, 8:10-13, 9:9-14; Lopresti ¶316.

A POSITA would understand that Hartono’s “task agents” (also called “application class agents”) correspond to the claimed *autonomous executable domain agent* in the ’825 Patent because they are software modules specific to an application or application area (meetings, email, personal information manager, video training, etc.). Lopresti ¶317; EX1006, 9:15-20 (“Each of the back-end application class agents 104a-d works with a class of one or more existing computer applications.”), 23:19-20 (“Each application class agent 104 preferably works with

a single class of applications 112 that have similar conceptual operations.”), 28:7-29 (Table IV listing “task agents”).

A POSITA would also understand that each of Hartono’s task agents are *configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands.* Lopresti ¶318. As Hartono explains:

Each task agent for a class of applications is preferably provided with its own set of semantically correct sentences, semantic actions and semantic mapping. Each task agent thus serves as the common user interface for the corresponding class of applications under the assumption that each application within the class accomplishes the same or generally overlapping sets of tasks.

EX1006, 27:17-28:3. To point, Table IV of Hartono provides a listing of different task agents and sample tasks in their *particular domain*. EX1006, Table IV. For example, the “mail” task agent has expertise in responding to commands to send, receive, compose and view email. EX1006, Table IV. The “address” task agent has expertise in managing address books. EX1006, Table IV; *see also* EX1006, 29:2-30:6. Accordingly, based on their domain of expertise, Hartono’s task agents each “include[] a process-specific execution module for each application.” EX1006, 30:15-20 (“For example, the word processing task agent may include an execution

module for MICROSOFT WORD”), 36:7-9 (“application agent...[is] adapted to output an executable instruction to an application program”); Lopresti ¶¶319-323.

Hartono further discloses using a “task interpretation engine” to *select at least one domain agent based on the determined domain*. Lopresti ¶¶324-325; EX1006, 24:21-25:3 (“The task interpretation engine 104B receives the semantic frame representation as input. Based on the frame’s head verb (action request) and noun phrases (parameters), the task interpretation engine 104B invokes a routine that sends a set of requests to the task application wrapper 104A.”), 15:13-15 (“A combination of the action-concept and the topic-concepts are used to determine which task agent should handle the request. If the request is for a specialized task agent, the request is routed to that specialized agent.”).

7. **5[f]: “providing...”**

Hartono discloses this limitation. Lopresti ¶¶326-328. Hartono discloses that its task agents are invoked by being *provided at least one query and/or command based on the natural language utterance*. Lopresti ¶¶327-328; EX1006, Abstract (“[A]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 4:20-25, 5:4-15, 36:7-9 (Hartono’s “application agent” is “adapted to receive and further interpret the output command

from the natural language agent and adapted to output an executable instruction to an application program.”).

8. **5[g]: “creating...”**

Hartono discloses this limitation. Lopresti ¶¶329-330. Hartono discloses that its task agents (*domain agents*) are invoked to output an executable instruction to an application program based on the received natural language command (*creating...one or more queries based on the at least one query and/or command*). EX1006, Abstract (“[A]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 4:20-25, 5:4-15, 5:21-25 (“[O]utputting the output command to an application class agent, semantically interpreting the output command and generating an executable instruction from the output command, and outputting the executable instruction to an application program for execution by the application.”), 36:7-9 (Hartono’s “application agent” is “adapted to receive and further interpret the output command from the natural language agent and adapted to output an executable instruction to an application program.”).

9. **5[h]: “sending...”**

Hartono discloses this limitation. Lopresti ¶¶331-336. Hartono discloses that its application task agents include agents (*at least one of the plurality of domain*

agents) with applications that perform Internet searches such as the “web” task agent, “Internet Product-Information lookup,” and “Internet Event look-up” agents. EX1006, Table IV, 39:1-13, 41:1-13.

As discussed for Ground 1, a POSITA would understand that the term “asynchronous” in the programming context includes sending the queries to be executed autonomously, rather than being controlled by the main programming flow. *See* Section VII.C.9 (Ground 1, 5[h]); Lopresti ¶332.

Hartono’s application class agents (*domain agent*) *send the one or more queries in an asynchronous manner to one or more local or external information sources*. Lopresti ¶¶333-336. First, Hartono discloses that its task agents (*domain agents*) operate autonomously (*see* limitation 5[e]) and thus send requests and/or commands *asynchronously* to local or external information sources apart from the main programming flow of recognizing and processing the speech input. *See* EX1006, 30:15-20 (“For example, the word processing task agent may include an execution module for MICROSOFT WORD”), 36:7-9 (“application agent...[is] adapted to output an executable instruction to an application program”). Second, Hartono’s application class agents “communicate[] with specific application 112 allowing the incorporation of existing applications into the architecture of the system,” via content-specific wrappers. EX1006, 24:4-13. Accordingly, a POSITA

would have understood that Hartono’s application task agents send the one or more queries to different types of “vendor-specific application programs” (*local or external information sources*). EX1006, 24:4-13; Lopresti ¶336. Moreover, Hartono discloses that these vendor-specific application programs would include web search engines over the Internet. EX1006, Table IV (detailing a “web” task agent,” with vendor-specific applications perform the tasks of “connect[ing], navigat[ing], search[ing] the internet/world-wide-web”), 26:1-6; Lopresti ¶336.

D. Dependent Claims

1. Claim 6: “...generating...”

Hartono discloses this limitation. Lopresti ¶¶338-340. Hartono discloses that the natural language syntactic parser and semantic interpreter generate (*generating, by the parser*) at least one operation based on the natural language input from the user (*at least one query and/or command based on the natural language utterance*). Lopresti ¶339. Hartono explains that “[t]he front end natural language 102 receives all natural language input and determines which of the available task agents 104 to pass the natural language input interpreted by the front end natural language agent 102.” EX1006, 10:14-17. As explained for 5[d] and 5[e], the syntactic parser and semantic interpreter determine and select which task agent (*domain agent*) to respond to the query. See Sections IX.C.5-6 (Ground 3, Limitations 5[d]-5[e]). The

syntactic parser and semantic interpreter also generates the operations as part of the process. *See* EX1006, 13:18-21 (“The natural language semantic interpreter 102c takes the syntactic parse of a spoken language request and maps it into a generic concept frame used to invoke the appropriate application method.”), Table I.

2. **Claim 7: “...comprises a plurality...”**

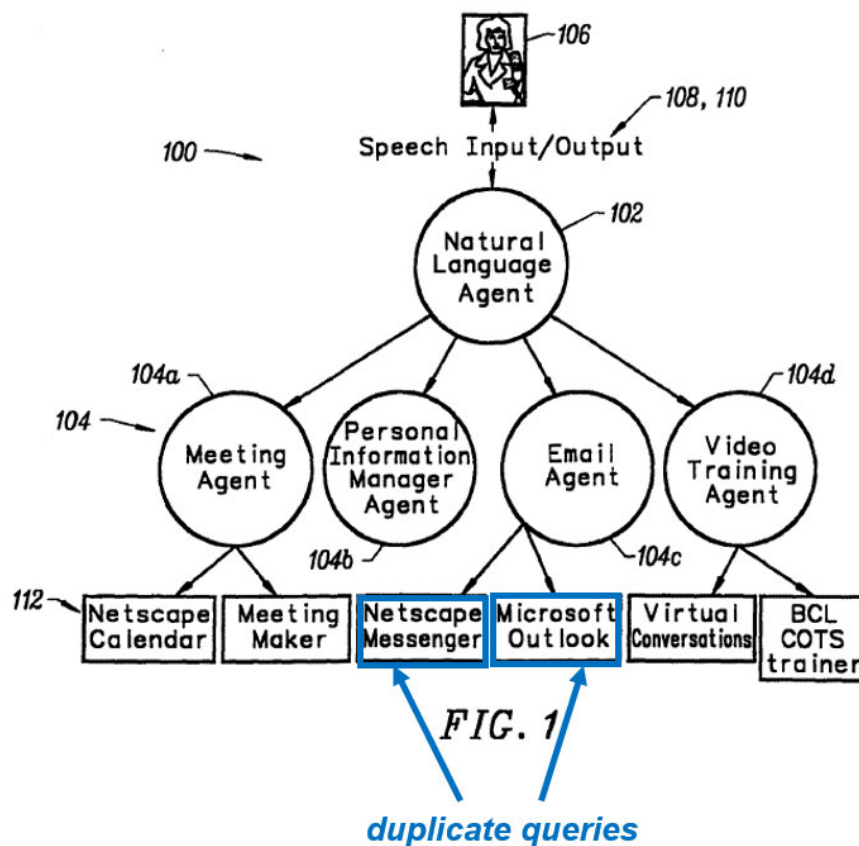
Hartono discloses this limitation. Lopresti ¶¶341-343. As discussed for claim 6, Hartono discloses that the natural language syntactic parser and semantic interpreter generate (*generating, by the parser*) at least one operation based on the natural language input from the user (*at least one query and/or command based on the natural language utterance*). *See* Section IX.D.1 (Ground 3, Claim 6). Hartono further discloses that the operation can include a set of requests for multiple different applications. Lopresti ¶342. Hartono explains that “[t]he communication between the application class agent 104 and each different type of vendor-specific applications programs 112 is via an application wrapper 104A that translates the conceptual action to *a set of application specific operations*.” EX1006, 24:4-7; *see also* 23:20-23 (“For example, different email applications generally perform the same conceptual actions of sending and receiving mail but performs these actions through different sets of steps.”).

3. **Claim 8: “...create...and send...”**

Hartono discloses this limitation. Lopresti ¶¶344-349. Like the ’825 Patent, Hartono discloses sending duplicate queries to different applications that perform the same task. *See* Section VIII.D.3 (Ground 2, Claim 8) (discussing the ’825 Patent’s description of *duplicate queries*); Lopresti ¶345.

Hartono discloses that the task agents are configured to translate an action to a set of application specific operations (*one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries*). Lopresti ¶347. As discussed for limitation 5[e], Hartono’s “task agents” (also referred in Hartono as “application class agents”) correspond to the claimed *domain agents*. *See* Section IX.C.6 (Ground 3, Limitation 5[e]). Hartono explains that “[t]he communication between the application class agent 104 and each different type of vendor-specific applications programs 112 is via an application wrapper 104A that translates the conceptual action to *a set of application specific operations*.” EX1006, 24:4-7. Hartono further discloses that these operations are *duplicates* because each operation executes the same query for different applications. *See* EX1006, 23:19-23 (“Each application class agent 104 preferably works with a single class of applications 112 that have similar conceptual operations”), Fig. 1 (below). “For example, an e-mail agent would have a wrapper for interacting with each email

system such as NETSCAPE and MICROSOFT EXCHANGE.” EX1006, 24:11-13; *see also* 26:25-3 (“An application wrapper interfaces with the particular address book application. The routines will handle the tasks as described above and will interface to the address book module for, e.g., MICROSOFT EXCHANGE, and NETSCAPE”).



EX1006, Fig. 1 (annotated)

Hartono further discloses that the tasks agents (*one or more of the selected at least one of the plurality of domain agents*) will send the set of application specific

operations (*send the duplicate queries*) to different particular applications, such as MICROSOFT EXCHANGE and NETSCAPE, independent of each other (*in an asynchronous manner to different information sources*). Lopresti ¶348. Hartono explains that “the application class agent 104 communicates with specific applications 112 allowing the incorporation of existing applications into the architecture of the system 100.” EX1006, 24:9-11. As discussed for limitations 5[e] and 5[h], the execution of these operations are *asynchronous* because they operate autonomously. *See* Sections IX.C.6 and IX.C.9 (Ground 3, Limitations 5[e] and 5[h]); *see also* EX1006, 23:20-23 (“For example, different email application generally perform the same conceptual actions of sending and receiving mail but performs these actions through different sets of steps.”).

X. GROUND 4

A. Motivation to Combine

A POSITA would have been motivated to add Glover’s metasearch functionality as a task agent to Hartono to aid in gathering relevant information from the Internet. Lopresti ¶¶351-361. As discussed for 5[h], Hartono discloses that its application task agents include agents with applications that perform Internet searches such as the “web” task agent, “Internet Product-Information lookup,” and “Internet Event look-up” agents. *See* Section IX.C.9 (Ground 3, 5[h]); EX1006, Table IV, 39:1-13, 41:1-13; Lopresti ¶352. Hartono thus already contemplates that

its application task agents (*domain agents*) would be creating and sending queries over the Internet via its vendor specific applications. Lopresti ¶353. Hartono does not detail how its task agents create and send queries over the Internet, and therefore, a POSITA would have been motivated to implement Hartono in a way to query multiple information sources over the Internet to obtain the most relevant results (such disclosed in Glover). Lopresti ¶¶354-355. This is especially true because Hartono’s application task agents like the “Internet Product-Information lookup” and “Internet Event look-up agents” that look to find specific information (e.g., about events, products, and prices), in which it would be desirable to obtain high-quality, relevant information. Lopresti ¶356.

A POSITA would look to implement in Hartono a metasearch engine like Glover that includes a dispatcher as a task agent that operates “similar to the query processor of a conventional search engine,” as it “generate[s] search engine requests from the user’s input,” to multiple search engines. EX1011, ¶38; Lopresti ¶357. A POSITA would have found a metasearch engine, like Glover, that acts as Hartono’s Internet-focused application task agents would have allowed for a broad search over several Internet databases to provide “improve[d] coverage compared to a single search engine.” EX1011, ¶35; Lopresti ¶358.

A POSITA would have been further motivated to implement Glover’s metasearch functionality because it uses “selective retrieval,” which “predicts the relevance of documents returned by regular search engines.” EX1011, ¶5; Lopresti ¶359. Specifically, Glover discloses a metasearch engine that performs continuous searching based on whether the returned results exceed a pre-determined confidence threshold. EX1011, ¶83. For those results below the threshold, Glover’s metasearch engine performs additional searches until it has confidence that the returned result provides relevant results. EX1011, ¶83. This “computation of the confidence of the relevance prediction,” can be done using “similarity measures..., or machine learning methods such as neural networks or support vector machines.” EX1011, ¶¶85-86. By adopting Glover’s “selective retrieval,” Hartono’s ability for its Internet-focused task agents to search the Internet and return results would benefit from “improve[d] execution time, resource usage, and throughput by requiring fewer retrieval requests compared to content-based metasearch engines.” EX1011, ¶5; Lopresti ¶¶360-361.

B. Reasonable Expectation of Success

A POSITA would have easily been able to implement a metasearch engine for use by Hartono and to allow for Glover’s dispatcher to act as Hartono’s application task agents to invoke Internet searching. Lopresti ¶¶362-364. Glover’s metasearch

engine simply extends Hartono's Internet search functionality. Lopresti ¶363. A POSITA would have the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and Internet searching based on user queries) according to known methods (speech recognition engines that convert speech into text strings and using text strings representing user's queries to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various applications to retrieve information from the Internet. Lopresti ¶364.

C. Claim 8: "...create...and send..."

Hartono in view of Glover renders obvious this limitation. Lopresti ¶¶366-371. In the combination, Hartono's Internet-focused task agents (*domain agents*) would have employed Glover's metasearch engine to create and send queries to different information sources based on the user's input. Lopresti ¶367. Specifically, Glover's dispatcher would act as a task agent (*domain agent*) to send queries to multiple search engines (*different information sources*). Lopresti ¶368

Glover discloses that as a general matter, "a metasearch engine takes user queries and submits them to multiple underlying search engines." EX1011, ¶35. It further discloses that its dispatcher to "choose search engines to query and...modify

queries appropriately for each source.” *Id.*, ¶39. A POSITA would have understood that taking a single user query and submitting to multiple search engines discloses *creating duplicate queries and sending the duplicate queries to different information sources*, as a query must be duplicated in order to be sent to multiple sources. Fig. 3 (showing “request submitter” sending duplicate queries to the World Wide Web); Lopresti ¶¶369-370. A POSITA would have also understood that Glover sends the queries to the World Wide Web (*external information sources*) *in an asynchronous manner*, as Glover states that its “steps may be performed in parallel,” “[f]or example, step 53 may be estimating the relevance of one or more results while step 52 is still sending queries 51 to and retrieving results 52 from search engines.” EX1011, ¶82; Lopresti ¶371.

XI. SECONDARY CONSIDERATIONS

There are no secondary considerations known to Petitioner that affect, let alone overcome, this strong case of obviousness. Should Patent Owner proffer any relevant evidence of secondary considerations, Petitioner reserves the right to address.

XII. MANDATORY NOTICES

A. Real Parties-in-Interest

The real party-in-interest is Meta Platforms, Inc.

B. Related Matters

U.S. Patent Office records indicate that the '825 Patent is assigned to Dialect, LLC ("PO"), which is currently asserting the '825 Patent in the following concurrent litigation filed on February 7, 2025: *Dialect, LLC v. Meta Platforms, Inc.*, 7:25-cv-00060 (W.D. Tex.).

There are five patents asserted the concurrent litigation. Petitioner has filed petitions for *inter partes* review against U.S. Patent No. 8,447,607 (IPR2025-01332) and U.S. Patent No. 9,263,039 (IPR2025-01333).

Petitioner has also filed, at substantially the same time that this Petition was filed, petitions for *inter partes* review against related family members asserted in the concurrent litigation: U.S. Patent No. 7,398,209 and U.S. Patent No. 8,015,006.

The '825 Patent is also asserted by PO in the following pending case: *Dialect, LLC v. Microsoft Corp.*, Case No. 2:24-vv-01067 (E.D. Tex.).

C. Notice of Counsel and Service Information

Petitioner provides the following designation of counsel:

Lead Counsel	Lisa K. Nguyen (Reg. No. 58,018) Paul Hastings LLP 1117 S. California Avenue Palo Alto, CA 94303 Telephone: 650.320.1800 Fax: 650.320.1900 Email: PH-META-DialectIPR@paulhastings.com
--------------	--

Back-Up Counsel	David Tennant (Reg. No. 48,362) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036 Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Diane Ghrist (<i>pro hac vice</i> to be filed) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036 Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Rachel Wu Hankinson (<i>pro hac vice</i> to be filed) 525 South Flower Street Twenty-Fifth Floor Los Angeles, CA 90071 Telephone: 213.683.6112 Fax: 213.627.0705 Email: PH-META-DialectIPR@paulhastings.com

Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this petition. Petitioner consents to electronic service by e-mail.

D. Fee for *Inter Partes* Review

The Director is authorized to charge the fee specified by 37 C.F.R. § 42.15(a) to Deposit Account 50-2613.

E. Grounds for Standing

Petitioner certifies pursuant to 37 C.F.R. § 42.104(a) that the '825 Patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

XIII. CONCLUSION

For these reasons, Petitioner respectfully requests institution.

Dated: August 15, 2025

Respectfully submitted,

By: / Lisa K. Nguyen/
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, LLC

CLAIM LISTING APPENDIX

5. A method for responding to a user generated natural language speech utterance, the method comprising:

[a] recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;

[b] receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;

[c] determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;

[d] determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;

[e] selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;

[f] providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;

[g] creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;

[h] sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.

6. The method of claim 5, further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.

7. The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.

8. The method of claim 5, wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that foregoing
PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,734,825
exclusive of the parts exempted as provided in 37 C.F.R. §42.24(a), contains 13,453
words and therefore complies with the type-volume limitations of 37 C.F.R.
§42.24(a).

Dated : August 15, 2025

Respectfully submitted,

By: / Lisa K. Nguyen /
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, INC.

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the foregoing **PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 9,734,825 AND EXHIBITS
EX1001-1026** are being served on August 15, 2025 via Federal Express overnight
mail on counsel of record for U.S. Patent No. 9,734,825 as addressed below:

**David A. Gerasimow
The Law Offices of David A. Gerasimow, P.C.
P.O. Box 10861
Chicago, IL 60610**

A courtesy copy is also being served on counsel for the patent holder in the
pending litigation:

Mark D. Siegmund (msiegmund@cjsjlaw.com)
CHERRY JOHNSON SIEGMUND JAMES, PC
Bridgeview Center
7901 Fish Pond Rd., 2nd Floor
Waco, TX 76710
Telephone: (254) 732-2242

Dated : August 15, 2025

Respectfully submitted,

By: / Lisa K. Nguyen/
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, INC.